

TUGAS AKHIR - KI141502

PEMODELAN PROSES BISNIS INTERLIBRARY LOAN STUDI KASUS PERPUSTAKAAN ITS

HENDRA DARMAWAN
NRP 05111440000073

Dosen Pembimbing I
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.

Dosen Pembimbing II
Adhatus Solichah Ahmadiyah, S. Kom., M.Sc.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

PEMODELAN PROSES BISNIS INTERLIBRARY LOAN STUDI KASUS PERPUSTAKAAN ITS

**HENDRA DARMAWAN
NRP 05111440000073**

**Dosen Pembimbing I
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.**

**Dosen Pembimbing II
Adhatus Solichah Ahmadiyah, S. Kom., M.Sc.**

**DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

MODELING OF BUSINESS PROCESS INTERLIBRARY LOAN CASE STUDY ITS LIBRARY

**HENDRA DARMAWAN
NRP 05111440000073**

**Supervisor I
Prof. Drs. Ec. Ir. Riyanarto Sarno, M.Sc., Ph.D.**

**Supervisor II
Adhatus Solichah Ahmadiyah, S. Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PEMODELAN PROSES BISNIS INTERLIBRARY LOAN STUDI KASUS PERPUSTAKAAN ITS

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Manajemen Informasi
Sistem Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh

HENDRA DARMAWAN

NRP. 05111440000073

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Prof. Drs. Ec. Ir. RIYANARTO SARNI

M.Sc., Ph.D.

NIP: 19590803 198601 1 001

ADHATUS SOLICHAH AHMADIYAH,

S.Kom., M.Sc.

NIP: 19850826 201504 2 002



(pembimbing 1)

(pembimbing 2)

SURABAYA

JULI, 2018

[Halaman ini sengaja dikosongkan]

PEMODELAN PROSES BISNIS INTERLIBRARY LOAN STUDI KASUS PERPUSTAKAAN ITS

Nama : Hendra Darmawan
NRP : 051114400073
Departemen : Informatika – FTIK
Dosen Pembimbing I : Prof. Drs. Ec. Ir. Riyanarto Sarno,
M.Sc., Ph.D.
Dosen Pembimbing II : Adhatus Solichah Ahmadiyah, S.Kom.,
M.Sc.

ABSTRAK

Interlibrary Loan adalah proses dimana suatu perpustakaan dapat melakukan permintaan peminjaman material dari atau menyediakan material untuk perpustakaan lain. Saat ini perpustakaan dan ruang baca di ITS belum memiliki SOP Interlibrary Loan, sehingga peminjaman antar perpustakaan dan ruang baca masih bersifat manual dan tidak tercatat. Untuk membuat SOP yang baik, SOP yang terbentuk perlu dianalisa menggunakan pemodelan proses bisnis. Kekurangan pemodelan proses bisnis saat ini adalah algoritma untuk mendeteksi invisible tasks dan non-free-choice yang ada (α) tidak di support secara langsung oleh pengembang ProM, sehingga penggunaannya masih belum dapat dilakukan. Algoritma alpha miner sendiri memiliki kekurangan, yaitu tidak dapat menyaring data anomali seperti algoritma heuristic.

Peneliti mengajukan tiga metode untuk mengatasi kekurangan pada metode pemodelan proses bisnis yang telah disebutkan sebelumnya. Metode pertama adalah penggabungan penemuan invisible task dan non-free-choice, sehingga invisible task dan non-free-choice bisa ditemukan dalam satu model proses bisnis. Metode kedua adalah penambahan invisible task pada event log secara eksplisit, sehingga event log dapat digunakan pada ProM. Kondisi ProM saat ini tidak dapat menemukan invisible task

dan non-free-choice secara bersamaan, diharapkan dengan menerapkan metode kedua ini invisible task dan non-free-choice dapat ditemukan dalam satu model proses bisnis di ProM. Metode ketiga adalah load data tanpa melakukan penggabungan node atau relasi, sehingga analisis yang membantu proses pemodelan dapat dilakukan. Menggunakan metode ketiga ini trace clustering dan penyaringan data dapat dilakukan sehingga memungkinkan graf yang terbentuk terbebas dari data yang mengandung anomali.

Tugas akhir ini dapat membuktikan bahwa metode pertama dapat menemukan invisible task dan non-free-choice secara bersamaan. Metode kedua dapat menjembatani kekurangan ProM dengan menambahkan invisible task pada event log. Metode ketiga dapat mengimplementasikan beberapa analisis seperti trace clustering dan pengelompokan relasi berdasarkan aktivitas yang dapat dikembangkan untuk mengimplementasikan pemodelan proses bisnis dengan menggunakan algoritma yang lebih rumit dibandingkan metode yang telah digunakan sebelumnya.

Kata kunci: *Control Flow Pattern, Neo4j, Trace Clustering, Graph Database, Data Filtering, Invisible Task, Non-Free-Choice.*

MODELING OF BUSINESS PROCESS INTERLIBRARY LOAN CASE STUDY ITS LIBRARY

Student Name : Hendra Darmawan
NRP : 05111440000073
Major : Informatics – FTIK
Supervisor I : Prof. Drs. Ec. Ir. Riyanarto Sarno,
M.Sc., Ph.D.
Supervisor II : Adhatus Solichah Ahmadiyah, S.Kom.,
M.Sc.

ABSTRACT

Interlibrary Loan is a process whereby a library may request loan requests from or provide materials for other libraries. Currently, libraries and roars read in ITS do not have Standard Operating Procedure for Interlibrary Loan, so lending between libraries and reading rooms is still manual and not recorded. To create a good SOP, the established SOP needs to be analyzed using business process modelling. The lack of current business process modelling is that the algorithm for detecting invisible tasks and non-free-choice (alpha \$) is not supported directly by the ProM developer, so its use is not yet possible. Alpha miner algorithm itself has a drawback, that is not able to filter anomalous data such as the heuristic algorithm.

The researcher proposed three methods to address the deficiencies in the business process modelling methods mentioned earlier. The first method involves the invisible and non-free-choice discovery so that the invisible task and non-free-choice can be found in a single business process model. The second method is the explicit addition of invisible tasks in the event log, so the event log can be used in ProM. The current state of ProM cannot discover both invisible task and non-free-choice simultaneously, it is hoped that by applying this second method invisible task and non-free-choice can be discovered in one business process model in ProM. The third

method is to load data without merging nodes or relationships, so an analysis that helps the modelling process can be done. Using this third method of trace clustering and data filtering can be done so as to allow the formed graph to be free of data containing anomalies.

This final project can prove that the first method can find invisible task and non-free-choice simultaneously. The second method bridges the shortcomings of ProM by adding invisible tasks to the event log. A third method can implement multiple analyzes such as trace clustering and activity-based grouping relationships that can be developed to implement business process modelling using more complicated algorithms than previously used methods.

Keywords: *Control Flow Pattern, Neo4j, Trace Clustering, Graph Database, Data Filtering, Invisible Task, Non Free Choice.*

KATA PENGANTAR

Segala puji syukur penulis kepada Allah SWT atas segala nikmat dan karunia-Nya, sehingga tugas akhir berjudul “Pemodelan Proses Bisnis *Interlibrary Loan* Studi Kasus Perpustakaan ITS” ini dapat selesai sesuai dengan waktu yang telah ditentukan.

Pengerjaan tugas akhir ini menjadi sebuah sarana untuk penulis memperdalam ilmu yang telah didapatkan di Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Teknik Informatika. terselesaikannya buku tugas akhir ini tidak terlepas dari bantuan dan dukungan semua pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Keluarga yang selalu memberikan dukungan berupa doa dan motivasi selama proses pengerjaan Tugas Akhir.
2. Bapak Riyanarto Sarno dan Ibu Adhatus Solichah selaku dosen pembimbing yang telah bersedia meluangkan waktu selama proses pengerjaan Tugas Akhir.
3. Teman-teman TC Angkatan 2014 terutama Rizal, Rey, dan Andy. Serta senior S2 maupun S3 terutama Cahya yang selalu mendukung selama proses pengerjaan tugas akhir.
4. Serta semua pihak yang turut membantu penulis dalam menyelesaikan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak sekali kekurangan. Dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan.

Surabaya, Juli 2018

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

LEMBAR PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER.....	xxiii
DAFTAR SINGKATAN.....	xxv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan	3
1.3 Batasan Permasalahan.....	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Metodologi.....	4
1.7 Sistematika Penulisan	6
BAB II DASAR TEORI	9
2.1 <i>Interlibrary Loan</i>	9
2.2 <i>Process Discovery</i>	9
2.3 <i>Event Log</i>	11
2.4 Neo4J	12
2.5 <i>Invisible Task</i>	13
2.6 Algoritma $\alpha++$ dan <i>Non-Free-Choice</i>	14

2.7	Pemodelan Proses Bisnis Menggunakan Neo4J	16
2.8	<i>Fitness dan Precision</i>	17
2.9	<i>Trace Clustering</i>	17
BAB III METODE PEMECAHAN MASALAH		19
3.1	Cakupan Permasalahan	19
3.2	Data.....	20
3.3	Pembentukan Business Process Model Menggunakan Neo4J	20
3.4	Metode Pertama	23
3.5	Metode Kedua.....	25
3.6	Metode Ketiga	30
3.7	Perbandingan Ketiga Metode.....	31
BAB IV IMPLEMENTASI		33
4.1	Lingkungan Implementasi	33
4.1.1.	Perangkat Keras.....	33
4.1.2.	Perangkat Lunak	33
4.2	Penjelasan Implementasi.....	34
4.2.1.	Implementasi Metode Pertama	34
4.2.2	Implementasi Metode Kedua	37
4.2.3	Implementasi Metode Ketiga	38
BAB V PENGUJIAN DAN EVALUASI.....		43
5.1	Lingkungan Uji Coba	43
5.2	Keluaran.....	43
5.2.1	Metode Pengujian	43
5.2.2	Skenario Uji Coba	44

5.3	Hasil Implementasi Pembentukan Model Bisnis Proses menggunakan Metode Pertama	44
5.3.1	Hasil Implementasi Pembentukan Model Bisnis Proses dari Event Log	44
5.3.2	Hasil Implementasi <i>Invisible Task</i> dan <i>Non-Free-Choice</i>	47
5.4	Hasil Implementasi Pembentukan Model Bisnis Proses Menggunakan Metode Kedua	51
5.5	Hasil Implementasi Pembentukan Model Bisnis Proses Menggunakan Metode Ketiga	55
5.5.1	Hasil Implementasi <i>Trace Clustering</i> , Penyaringan Data Anomali, dan Penggabungan <i>Node</i>	55
5.5.2	Hasil Penemuan <i>Invisible Task</i> dan <i>Non-Free-Choice</i>	59
5.6	Perbandingan Hasil Pemodelan Proses Bisnis	63
	BAB VI KESIMPULAN DAN SARAN	71
6.1	Kesimpulan	71
6.2	Saran	72
	DAFTAR PUSTAKA.....	73
	LAMPIRAN	75
	DAFTAR ISTILAH	99
	BIODATA PENULIS	101

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Contoh <i>Process Discovery</i>	11
Gambar 2.2 Definisi <i>Event Log</i>	12
Gambar 2.3 Contoh <i>Node</i> dalam <i>Neo4j</i>	13
Gambar 2.4 Contoh <i>Relationship</i> dalam <i>Neo4j</i>	13
Gambar 2.5 Contoh Jenis <i>Prime Invisible Task</i>	14
Gambar 2.6 Perbandingan 3 Hasil <i>Mining</i> antara Algoritma $\alpha++$ dengan Algoritma α atau $\alpha+$	15
Gambar 2.7 Contoh Hasil Penemuan <i>Non-Free-Choice</i>	16
Gambar 3.1 Alur Pengerjaan Tugas Akhir	19
Gambar 3.2 <i>Import Event Log</i> Menggunakan <i>Cypher Query</i> <i>Language</i>	21
Gambar 3.3 Alur Pengerjaan Metode Pertama	23
Gambar 3.4 Alur Pengerjaan Metode Kedua	25
Gambar 3.5 Contoh Penambahan <i>Invisible Task</i> pada <i>Event Log</i> dengan Tipe Berpindah	27
Gambar 3.6 Contoh Penambahan <i>Invisible Task</i> pada <i>Event Log</i> dengan Tipe Mengulang	28
Gambar 3.7 Contoh Penambahan <i>Invisible Task</i> pada <i>Event Log</i> dengan Tipe Melewati	29
Gambar 3.8 Alur Pengerjaan Metode Ketiga	30
Gambar 5.1 Hasil Penemuan <i>Sequence Event Log</i> pada Graf Neo4J	45
Gambar 5.2 Hasil Penemuan <i>Control-Flow Pattern</i> pada Graf Neo4J	47
Gambar 5.3 Hasil Penemuan <i>Invisible Task</i> pada Graf Neo4j	48
Gambar 5.4 Hasil Penemuan <i>Non-Free-Choice</i> pada Graf Neo4j	49
Gambar 5.5 Hasil Pemodelan Proses Bisnis dengan <i>Event Log</i> yang ditambahkan <i>Invisible Task</i> pada Graf Neo4J	53
Gambar 5.6 Hasil Pemodelan Proses Bisnis Tanpa Penggabungan <i>Node</i> dengan CaseId PP1 dan PP2	55
Gambar 5.7 Hasil <i>Trace Clustering</i> Menggunakan Neo4J	55
Gambar 5.8 Hasil Penyaringan Data dan Penemuan <i>Sequence</i> <i>Event Log</i> pada Graf Neo4J	57

Gambar 5.9 Hasil Penghitungan Panjang <i>Trace</i> dalam Setiap <i>Case</i> Menggunakan Neo4J	56
Gambar 5.10 Hasil Penemuan Control-Flow Pattern pada Graf Neo4J	59
Gambar 5.11 Hasil Pengelompokan Relasi <i>Sequence</i> Tiap Aktivitas Menggunakan Neo4J.....	60
Gambar 5.12 Hasil Penemuan <i>Non-Free-Choice</i> pada Graf Neo4J	61
Gambar 5.13 Hasil Pemodelan Proses Bisnis <i>Interlibrary Loan</i> Menggunakan ProM	65
Gambar 5.14 Hasil Pemodelan Proses Bisnis <i>Interlibrary Loan</i> dengan <i>Event Log</i> yang Mengandung <i>Invisible Task</i> Menggunakan ProM	65
Gambar 5.15 Hasil Pemodelan Proses Bisnis <i>Interlibrary Loan</i> dengan <i>Event Log</i> yang telah disaring Menggunakan ProM	66
Gambar 5.16 Hasil Pemodelan Proses Bisnis yang Mengandung <i>Non-Free-Choice</i> Menggunakan ProM	66
Gambar 5.17 Hasil Pemodelan Proses Bisnis yang Mengandung <i>Non-Free-Choice</i> Menggunakan Neo4J	67
Gambar 5.18 Hasil Pemodelan Proses Bisnis dengan <i>Event Log</i> yang Mengandung <i>Invisible Task</i> dan <i>Non-Free-Choice</i> Menggunakan ProM	67
Gambar 5.19 Hasil Pemodelan Proses Bisnis dengan <i>Event Log</i> yang Mengandung <i>Invisible Task</i> dan <i>Non-Free-Choice</i> Menggunakan Neo4J	68
Gambar 5.20 Hasil <i>Fitness</i> dan <i>Precision</i> untuk Setiap Metode ..	69
Gambar 7.1 Model Proses Bisnis <i>Sequence</i>	94
Gambar 7.2 Model Prroses Bisnis <i>Sequence</i> dengan CaseId PP1 dan PP2	95
Gambar 7.3 Model Proses Bisnis Penemuan <i>Sequence Case</i>	97

DAFTAR TABEL

Tabel 3.1 Contoh <i>Event Log Interlibrary Loan</i>	20
Tabel 3.2 Algoritma Penemuan dalam Neo4J	22
Tabel 3.3 Algoritma yang digunakan pada Metode Pertama.....	24
Tabel 3.4 Algoritma yang digunakan pada Metode Kedua	26
Tabel 3.5 Algoritma yang digunakan pada Metode Ketiga	31
Tabel 5.1 <i>Event Log</i> Sebelum (Kiri) dan Setelah (Kanan) ditambahkan <i>Invisible Task</i>	51
Tabel 7.1 Hasil Trace Clustering Menggunakan Neo4J	75
Tabel 7.2 Hasil Pengelompokan Relasi <i>Sequence</i> Tiap Aktivitas	77
Tabel 7.3 Hasil Penghitungan Panjang <i>Trace</i> dalam Setiap Case	78
Tabel 7.4 <i>Event Log Interlibrary Loan</i>	81
Tabel 7.5 <i>Event Log Interlibrary Loan</i> yang Mengandung <i>Invisible Task</i>	86

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kueri Cypher 4.1 Mengubah <i>Event Log</i> ke <i>Link List</i>	34
Kueri Cypher 4.2 Penemuan XOR di Neo4J	35
Kueri Cypher 4.3 Penemuan AND di Neo4J	35
Kueri Cypher 4.4 Penemuan OR di Neo4J	35
Kueri Cypher 4.5 Penemuan <i>Invisible Task</i>	36
Kueri Cypher 4.6 Pembentukan Model Proses Bisnis <i>Sequence</i> dan Penemuan <i>Non-Free-Choice</i>	36
Kueri Cypher 4.7 Penemuan <i>Control-Flow Pattern XOR</i> pada <i>Event Log</i> yang Mengandung <i>Invisible Task</i>	37
Kueri Cypher 4.8 Penemuan <i>Non-Free-Choice</i> pada <i>Event Log</i> yang Mengandung <i>Invisible Task</i>	38
Kueri Cypher 4.9 Mengubah <i>Event Log</i> ke <i>Link List</i> dengan dipisahkan Setiap <i>Case</i>	39
Kueri Cypher 4.10 <i>Trace Clustering</i> dan Penghitungan Frekuensi <i>Case</i>	39
Kueri Cypher 4.11 Penghapusan Data Anomali	40
Kueri Cypher 4.12 Penggabungan <i>Node</i>	40
Kueri Cypher 4.13 Penghapusan Relasi NEXT	41
Kueri Cypher 4.14 Penghapusan <i>CaseId</i>	41

[Halaman ini sengaja dikosongkan]

DAFTAR SINGKATAN

csv	: <i>Comma Separated Value.</i>
SOP	: <i>Standard Operating Procedure</i>
ILL	: <i>Interlibrary Loan</i>

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pengerjaan Tugas Akhir, dan sistematika penulisan Tugas Akhir.

1.1 Latar Belakang

Setiap organisasi saat ini dihadapkan dengan banyaknya tantangan, seperti perubahan iklim perekonomian, persaingan bisnis, inovasi, dan penerapan teknologi baru. Organisasi harus mempunyai strategi dalam usaha merebut perhatian pasar. Suatu organisasi dituntut untuk dapat memenuhi keinginan konsumen dalam hal kualitas dan waktu yang tepat. Sehingga suatu organisasi harus memiliki Standar Operasional Prosedur (SOP) yang digunakan sebagai acuan dalam melaksanakan aktivitas.

Penelitian ini membahas mengenai penetapan SOP bagi perpustakaan pusat Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Diketahui bahwa perpustakaan pusat ITS belum menerapkan sistem *Interlibrary loan*, yaitu proses suatu perpustakaan melakukan permintaan peminjaman material dari atau untuk perpustakaan lain. *Interlibrary loan* ini merupakan salah satu bentuk *resource sharing* yang memiliki dua proses yaitu proses peminjaman dan proses penyediaan [1]. Sistem ini melibatkan banyak perpustakaan dan umumnya berbentuk website, sehingga untuk memudahkan transaksi, maka dibutuhkan SOP yang jelas. SOP ini harus tertulis pada website dan berisi tentang peraturan, kebijakan, dan batasan yang sudah ditentukan perpustakaan. Sehingga penting untuk melakukan identifikasi proses bisnis dan memodelkan supaya terbentuk SOP yang jelas. Adapun pembentukan model proses ini dengan graf database. Salah satu perangkat lunak graf database adalah Neo4J.

Graf database berguna dalam memahami keragaman dataset dalam berbagai bidang. Graf database menyediakan teknik

pemodelan data yang kuat dan baru. Saat ini, suatu organisasi tidak hanya perlu mengelola volume data yang lebih besar, tetapi juga menggali informasi dari data yang ada. Sehingga, relasi antar data menjadi lebih penting. Untuk meningkatkan relasi data, maka teknologi database diperlukan untuk menyimpan informasi hubungan antardata. Tidak hanya menyimpan relasi data secara efektif, graf *database* juga fleksibel dalam menyesuaikan perubahan kebutuhan bisnis [2].

Pemodelan ini diawali dengan memperoleh *log* yang berisi informasi mengenai proses bisnis yang berjalan [3]. Nantinya *event log* ini akan dibentuk proses modelnya secara otomatis menggunakan suatu algoritma. Algoritma pemodelan yang umum digunakan adalah *alpha miner* dan iterasinya, yaitu algoritma *alpha*, *alpha+* [4], *alpha++* [5], dan *alpha#* [6]. Beberapa algoritma ini akan dijelaskan pada bab selanjutnya. Dalam pemodelan proses bisnis ada beberapa permasalahan yang ditemukan, seperti penemuan *invisible task* yang telah diselesaikan dengan algoritma *alpha#* dan penemuan *non-free-choice* yang telah diselesaikan dengan algoritma *alpha++*. Saat ini telah ada algoritma baru yang dapat mendeteksi kedua permasalahan diatas, yaitu algoritma *alpha\$*. Tetapi algoritma *alpha\$* [7] masih belum dapat digunakan untuk saat ini karena algoritma ini belum secara resmi di dukung secara resmi oleh pengembang program ProM sehingga pengguna secara luas masih belum dapat memecahkan kedua permasalahan diatas, ProM sendiri adalah program yang umumnya digunakan untuk melakukan analisa terhadap proses bisnis. Kekurangan selanjutnya adalah algoritma *alpha* tidak ada yang mengimplementasikan penyaringan data *event log* yang mengandung anomali. Anomali data dapat berupa data yang terlalu pendek atau terlalu panjang apabila dibandingkan dengan data lainnya, atau data yang memiliki frekuensi pemanggilan sangat sedikit.

Berdasarkan beberapa kekurangan tersebut, perlu ditemukannya metode baru untuk melakukan pemodelan proses bisnis yang dapat menemukan atau mendeteksi *invisible task* dan

non-free-choice, serta melakukan penyaringan data terhadap event log. Penulis mengusulkan tiga metode untuk menyelesaikan kekurangan-kekurangan metode penemuan proses bisnis yang ada sekarang dengan menggunakan graf *database* Neo4J. Metode pertama adalah menemukan *invisible task* dan *non-free-choice* dalam satu model. Metode kedua yaitu menambahkan *invisible task* pada event log, metode ini berguna ketika pengguna ingin melakukan analisa menggunakan ProM karena ProM saat ini tidak bisa mendeteksi *invisible task* dan *non-free-choice* pada model proses bisnis dalam satu model. Metode ketiga adalah menggunakan *load* atau *import* data tanpa menggunakan penggabungan node dan relasi, metode ini memiliki banyak fungsi seperti melihat jumlah relasi antar aktivitas, *trace clustering*, dan penyaringan data. *Trace clustering* sendiri adalah pengelompokan kasus atau *case* berdasarkan aktivitas apa saja yang ada di dalam *case* tersebut. Diharapkan metode pertama dapat menemukan *invisible task* dan *non-free-choice* dalam satu model, metode kedua dapat menjembatani pemodelan proses bisnis antara Neo4J dengan ProM, dan metode ketiga dapat mengatasi kekurangan pada algoritma alpha miner yang tidak dapat melakukan penyaringan data.

1.2 Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana cara membuat model proses bisnis yang benar menggunakan graf database Neo4J?
2. Bagaimana cara membuat model proses bisnis yang benar dan dapat digunakan pada Neo4J maupun ProM?
3. Bagaimana cara untuk melakukan penyaringan data yang mengandung anomali?

1.3 Batasan Permasalahan

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, diantaranya sebagai berikut:

1. Model proses bisnis diimplementasikan pada graf *database* Neo4j.
2. Data masukan berupa *event log* dalam bentuk Excel.
3. Konversi data dari *file* excel ke mxml menggunakan program Disco.
4. ProM 5.2 digunakan untuk mengecek kebenaran hasil penemuan.

1.4 Tujuan

Tujuan dari Tugas Akhir ini diharapkan dapat membuat SOP Interlibrary Loan yang dapat diterapkan dalam ruang baca maupun Perpustakaan ITS dan metode pemodelan proses bisnis yang dapat mengatasi kekurangan pada metode yang telah ada sebelumnya.

1.5 Manfaat

Manfaat dari penelitian ini antara lain:

1. Memberikan rekomendasi *Standard Operating Procedure* kepada Perpustakaan ITS untuk membuat sistem *Interlibrary Loan*.
2. Metode yang digunakan pada Tugas Akhir ini dapat dikembangkan lebih lanjut dengan algoritma yang lebih kompleks menggunakan Neo4J.
3. *Trace clustering* dan penyaringan data yang mengandung anomali dapat digunakan dan dikembangkan oleh individu atau organisasi yang menggunakan Neo4J untuk menggambarkan proses bisnis.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu sebagai berikut:

1. Studi literatur

Dalam pembuatan Tugas Akhir ini telah dipelajari tentang hal-hal yang dibutuhkan sebagai ilmu penunjang dalam penyelesaiannya. Ilmu penunjang utama pada Tugas Akhir ini adalah permodelan menggunakan Neo4J.

Selain itu, terdapat literatur lain yang menunjang proses penyelesaian Tugas Akhir ini.

2. Pembuatan Metode

Pada tahap ini penulis menjabarkan cara pemecahan masalah yang terdapat dalam rumusan masalah.

3. Analisis dan Perancangan Algoritma

Tahap ini meliputi perancangan algoritma berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini.

4. Implementasi

Pada tahap ini dilakukan penjelasan terhadap kueri Cypher yang akan digunakan pada tahap Pengujian dan Evaluasi.

5. Pengujian dan evaluasi

Pada tahap ini dilakukan implementasi pemodelan pada neo4j yang menghasilkan tiga graf, graf pertama terbentuk dari hasil input langsung dari *event log*, graf kedua terbentuk dari event log yang telah ditambahkan *invisible task*, graf ketiga terbentuk dari data yang sebelumnya telah disaring. Kemudian dilakukan perbandingan terhadap ketiga graf tersebut. Tahapan-tahapan dari implementasi adalah sebagai berikut:

- a. Load event log dengan atau tanpa *invisible task* yang telah dibuat kedalam neo4j.
- b. Tambahkan relasi sequence dan *control-flow pattern* kedalam graf.
- c. Penemuan *invisible task* dan *non-free-choice* ke dalam graf tanpa *invisible task*. Penemuan *non-free-choice* ke dalam graf yang *event log* berisi *invisible task*.
- d. Melakukan trace clustering pada proses bisnis.

- e. Menyaring data yang termasuk sebagai data anomali.
- f. Membandingkan ketiga model proses bisnis yang di *discovery* dengan metode pemodelan yang langsung membaca dari *event log* tanpa modifikasi, Model proses bisnis dengan *event log* yang mengandung *invisible task*, dan model proses bisnis yang di *discovery* setelah melakukan penyaringan data yang mengandung anomali.

6. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan yang menjadi dasar dari pembuatan Tugas Akhir ini.

Bab III Metode Pemecahan Masalah

Bab ini berisi tentang pembahasan masalah, perencanaan pemecahan masalah, serta metode yang akan diimplementasikan pada bab V.

Bab IV Implementasi

Bab ini berisi tentang kueri Cypher yang akan digunakan untuk pemodelan proses bisnis menggunakan Neo4J.

Bab V Pengujian dan Evaluasi

Bab ini membahas hasil dari metode yang telah di jelaskan di bab III dan hasil dari implementasi kueri cypher pada bab IV.

Bab VI Kesimpulan

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

Lampiran

Merupakan lampiran yang digunakan untuk menampilkan data atau gambar yang lebih lengkap dari bab-bab selanjutnya.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir.

2.1 *Interlibrary Loan*

Berdasarkan [1] *Interlibrary Loan* (ILL) adalah proses dimana suatu perpustakaan dapat melakukan permintaan peminjaman material dari atau menyediakan material untuk perpustakaan lain. ILL merupakan salah satu bentuk *resource sharing*. Dapat disimpulkan dari penjelasan diatas bahwa proses ILL dapat dibagi menjadi dua yaitu proses peminjaman atau *request* dan proses penyediaan atau *supply*, karena ILL melibatkan banyak perpustakaan dan umumnya dalam bentuk website. Untuk memudahkan transaksi dalam kasus ini adalah peminjaman maka dibutuhkan *Standard Operating Procedure* (SOP) yang jelas. SOP harus tertulis pada website ILL dan berisi tentang peraturan, kebijakan, atau batasan yang sudah ditentukan oleh perpustakaan atau institusi yang berkait untuk diketahui oleh pengguna.

Pengguna bertindak sebagai pemustaka, melakukan permintaan peminjaman terhadap material yang akan dipinjam, tetapi material yang akan dipinjam harus sudah dipastikan tidak ada di dalam perpustakaan tempat dimana pemustaka akan melakukan permintaan material oleh pustakawan. Material apa saja yang bisa dipinjam oleh pengguna juga harus ditentukan dan dicantumkan pada website ILL. Hal ini merupakan salah satu contoh SOP pada ILL. SOP diambil dari beberapa sumber [1] [8].

2.2 *Process Discovery*

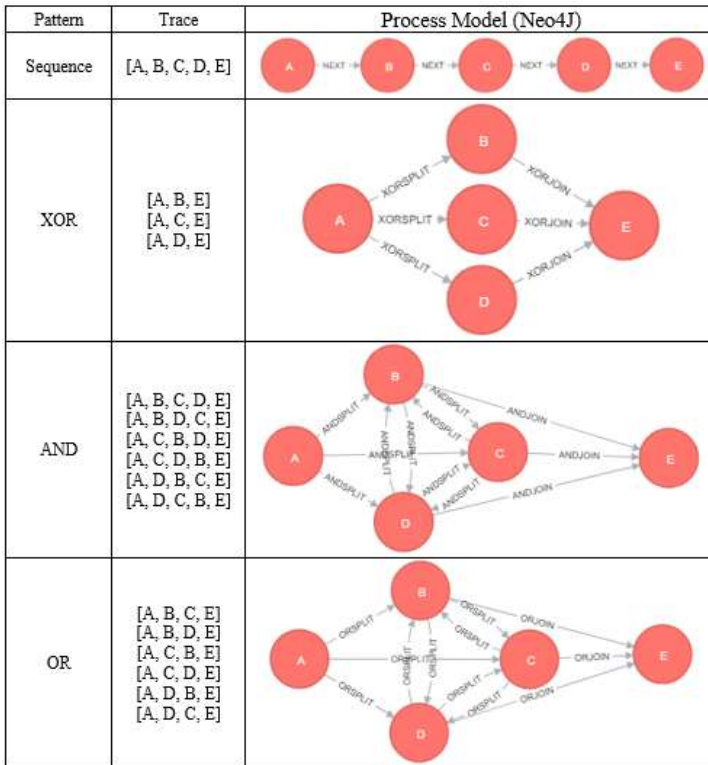
Proses mining adalah suatu ilmu yang menggabungkan antara pembelajaran mesin serta *data mining* pada satu sisi dan memodelkan serta menganalisa proses pada sisi yang lain. Ide dasar dari *process mining* adalah menemukan, memantau serta

meningkatkan proses sebenarnya dengan mengambil pengetahuan dari *event log* yang ada pada suatu sistem [3].

Proses *mining* dibagi menjadi tiga bagian, salah satunya adalah *process discovery*. *Process discovery* adalah suatu teknik yang memanfaatkan *event log* dalam penentuan proses model tanpa menggunakan informasi-informasi lainnya. *Process discovery* yang ada diantaranya algoritma *Alpha#* [9] serta *Heuristics Miner* [10].

Proses model yang benar adalah proses model yang mampu menampilkan konkurensi serta *control-flow pattern* [3]. Konkurensi berarti aktivitas pada proses model tidak ditampilkan secara redundan (tidak ada aktivitas yang sama ditampilkan pada proses model) [3]. *Control-flow pattern* adalah penanda relasi antar aktivitas. Terdapat empat relasi antar aktivitas yaitu relasi *sequence*, relasi XOR, relasi AND and relasi OR [3].

Relasi *sequence* adalah relasi yang terjadi untuk menghubungkan satu aktivitas dengan satu aktivitas lainnya. Relasi XOR adalah relasi yang terjadi yang hanya memperbolehkan salah satu aktivitas yang terjadi pada suatu proses apabila ada cabang pada aktivitas tersebut. Relasi OR adalah relasi yang terjadi apabila ada beberapa aktivitas pada satu cabang dan pengguna diharuskan melakukan beberapa aktivitas dalam cabang tersebut. Sedangkan relasi AND adalah relasi yang terjadi dimana semua aktivitas pada satu percabangan harus dijalankan semuanya. Apabila aktivitas pilihan pada relasi XOR, OR dan AND adalah aktivitas sebelum dari aktivitas lainnya, maka *control-flow pattern* dari relasi tersebut adalah penambahan kata “Split” pada relasi tersebut [3]. Sedangkan, apabila aktivitas pilihan pada relasi XOR, OR dan AND adalah aktivitas setelah dari aktivitas lainnya, maka *control-flow pattern* dari relasi tersebut adalah penambahan kata “Join” pada relasi tersebut [3]. Gambar 2.1 merupakan contoh penggunaan relasi pada proses model, dapat dilihat pada gambar Gambar 2.1 suatu proses bisnis harus dapat menggambarkan beberapa *control-flow pattern*.

Gambar 2.1 Contoh *Process Discovery*

2.3 Event Log

Event log merupakan catatan kejadian dari suatu proses bisnis yang berjalan [3]. Sebuah kejadian menunjukkan aktivitas ataupun langkah dari suatu proses. *Event log* menunjukkan proses tunggal sehingga sebuah kejadian termasuk dalam sebuah proses, yang biasa disebut dengan *case* [3]. *Event log* dapat menyimpan informasi yang berhubungan dengan kejadian, seperti *timestamp* atau *resources* [3]. *Timestamp* adalah waktu terjadinya suatu kejadian. Sedangkan *actor* adalah pelaku yang mengeksekusi suatu kejadian. Definisi lengkap dari *event log* dapat dilihat pada Gambar 2.2 [3].

Definisi Event log

Event log terdiri dari rangkaian aktivitas dan waktu yang didefinisikan dengan $L_{A,TD} = (E, C, \alpha, \gamma, \beta, >)$ dimana:

- E adalah kumpulan kejadian.
- C adalah kumpulan *case*.
- $\alpha: E \rightarrow A$ adalah fungsi yang menghubungkan setiap kejadian dengan sebuah aktifitas.
- $\gamma: E \rightarrow TD$ adalah fungsi yang menghubungkan setiap kejadian dengan sebuah waktu kejadian (*timestamp*).
- $\beta: E \rightarrow C$ adalah fungsi yang menghubungkan sebuah kejadian dengan sebuah *case*.
- $> \subseteq E \times E$ adalah *relation succession*, yang merupakan total suatu kejadian yang dilanjutkan dalam kejadian lain dan termasuk dalam E. $e_2 > e_1$ adalah notasi singkat untuk $(e_2, e_1) \in >$. Kumpulan kejadian yang berkaitan dalam sebuah *case* disebut sebagai *trace*.

Gambar 2.2 Definisi Event Log

2.4 Neo4J

Neo4J adalah aplikasi gratis NoSQL graf database yang di implementasikan di java yang dibuat mulai 2003, dan telah tersedia untuk umum sejak 2007. Neo4j sekarang digunakan oleh ratusan ribu perusahaan dan organisasi di hampir semua industri. *Use case* yang meliputi *matchmaking*, manajemen jaringan, analisis perangkat lunak, penelitian ilmiah, *routing*, manajemen organisasi dan proyek, rekomendasi, jejaring sosial dan lainnya. Neo4j menerapkan model grafik properti secara efisien dalam tingkat penyimpanan.

Neo4j dapat menyimpan 2 hal yang terdiri dari: *Node Label* dan *Relationship Types*. Node merupakan kumpulan data yang berisi informasi-informasi yang tersimpan dalam sebuah data. Contoh: Sebuah database Person yang menampung informasi seperti nama, asal, umur, dsb. Sedangkan Relationship merupakan kumpulan data yang berisi hubungan atau relasi antar node. Contoh: dari Gambar 2.3 dibuat sebuah relasi yang menyatakan Hendra KENAL Darmawan. Gambar 2.4 menunjukan relasi tersebut.



Gambar 2.3 Contoh Node dalam Neo4j



Gambar 2.4 Contoh Relationship dalam Neo4j

2.5 Invisible Task

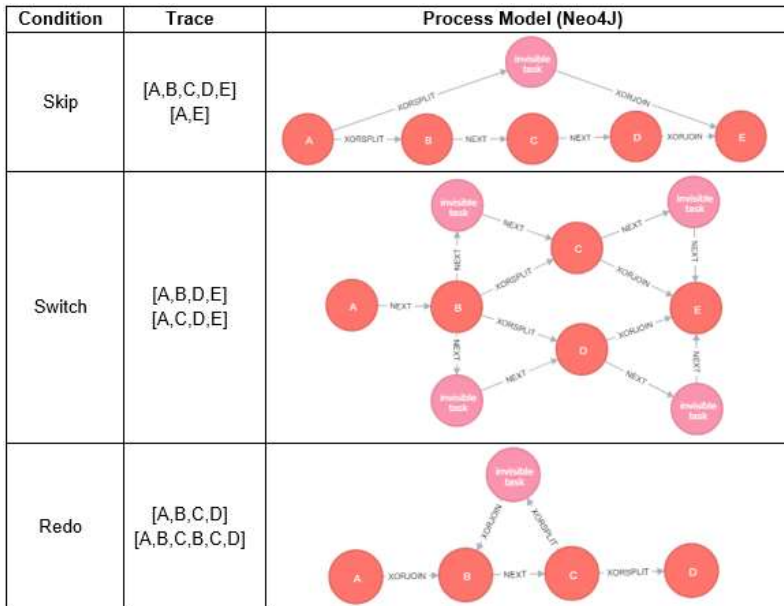
Invisible Task (Aktivitas Tak Terlihat) adalah aktivitas tambahan yang tidak muncul di *event log* tetapi ditampilkan dalam model [6]. Kegunaan dari *invisible task* adalah untuk membantu menggambarkan proses secara sebenarnya dalam model proses. *Invisible task* dibagi menjadi dua macam, yaitu *invisible prime task* dan *invisible non-prime task*. *Invisible prime task* adalah *invisible task* yang ditangani oleh algoritma Alpha# [6]. Terdapat tiga macam *invisible task prime* yaitu Melewati (*Skip*), Mengulang (*Redo*), dan Berpindah (*Switch*).

Invisible task tipe Melewati digunakan untuk menggambarkan aktivitas yang dapat dilewati. Pada Gambar 2.5, *trace* yang terdapat pada contoh tipe *skip* ada dua *trace*, yaitu [A, B, C, D, E] dan [A, E]. Pada *trace* kedua, aktivitas A tidak melewati aktivitas B, C, dan D melainkan langsung menuju ke aktivitas E. Hal ini yang menyebabkan *invisible task* dengan tipe Melewati perlu digambarkan dari aktivitas A ke aktivitas E.

Invisible task tipe Mengulang digunakan untuk menggambarkan aktivitas yang dapat diulang. Pada Gambar 2.5, *trace* yang terdapat pada contoh tipe *redo* ada dua *trace*, yaitu [A, B, C, D] dan [A, B, C, B, C, D]. Pada *trace* kedua, aktivitas B dan C diulang sebanyak sekali, maka *invisible task* perlu digambarkan diantara aktivitas C dan aktivitas B.

Invisible task tipe Berpindah digunakan untuk perpindahan eksekusi pada beberapa percabangan. Pada Gambar 2.5, *trace* yang

terdapat pada contoh tipe *switch* ada dua, yaitu [A, B, D, E] dan [A, C, D, E] yang membuat adanya percabangan atau pilihan antara aktivitas A dengan C, A dengan B, B dengan D, dan C dengan D. Hal ini menyebabkan *invisible task* perlu digambarkan di percabangan, baik percabangan dengan tipe split (membuka cabang) maupun dengan tipe join (menutup cabang).

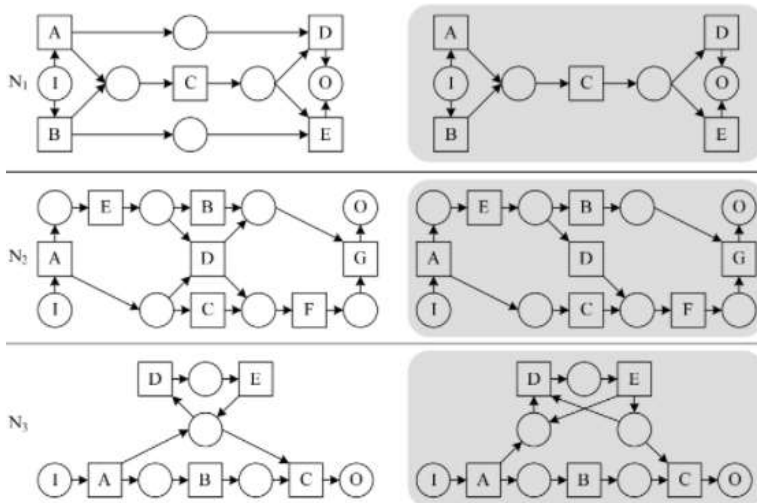


Gambar 2.5 Contoh Jenis *Prime Invisible Task*

2.6 Algoritma $\alpha++$ dan *Non-Free-Choice*

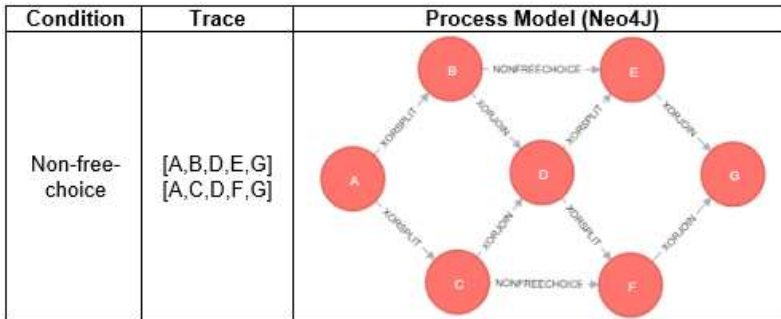
Algoritma $\alpha++$ merupakan perbaikan atau pengembangan dari algoritma α dan $\alpha+$. Algoritma α merupakan algoritma dasar yang dapat digunakan untuk menganalisa dan menghasilkan *activity* dari *case*. Kekurangan dari algoritma α adalah *length one loop*, *length two loop*, *invisible task*, *duplicate task*, *implicit places*, dan *non-free choice*. Perbaikan atau pengembangan dari algoritma α adalah algoritma $\alpha+$. Kelebihan algoritma $\alpha+$ dibandingkan

algoritma α adalah algoritma $\alpha+$ dapat melakukan *mining* terhadap *short loops* seperti length one loop yang tidak terdeteksi pada algoritma α . Kemudian $\alpha++$ dapat menangani *implicit dependency* yang mungkin terjadi pada sebuah *event log* yang telah dijalankan. *Implicit Dependency* dapat disebut juga sebagai *Indirect Dependency* merefleksikan hubungan tidak langsung antara 2 task atau activity yang saling berhubungan. Penjelasan rule algoritma $\alpha++$ terdapat pada [5].



Gambar 2.6 Perbandingan 3 Hasil Mining antara Algoritma $\alpha++$ dengan Algoritma α atau $\alpha+$

Gambar 2.6 adalah salah satu contoh perbandingan hasil mining dari algoritma $\alpha++$ dan algoritma α atau $\alpha+$. Model proses di sebelah kiri adalah hasil mining yang benar dari algoritma $\alpha++$ sedangkan gambar di sebelah kanan adalah hasil mining yang salah dari algoritma α atau $\alpha+$. Salah satu contoh dari *implicit dependency* pada gambar 1 model proses 1 A dan D, B dan E adalah *implicit dependency*. Terlihat jelas pada model proses 1 hasil algoritma pada sebelah kanan tidak dapat mendeteksi *implicit dependency* atau *non-free-choice* seperti algoritma $\alpha++$ di sebelah kiri.



Gambar 2.7 Contoh Hasil Penemuan Non-Free-Choice

2.7 Pemodelan Proses Bisnis Menggunakan Neo4J

Pemodelan proses bisnis menggunakan Neo4J telah dilakukan pada [11] dan [12]. Metode pemodelan proses bisnis menggunakan Neo4J dapat dilakukan dengan *load event log* ke dalam Neo4J. *Event log* di *load* dua kali dan dimasukkan ke label atau tabel yang berbeda. Setelah melakukan proses *load event log*, tahapan berikutnya adalah melakukan penemuan *sequence* sehingga *node-node* yang awalnya terpisah terbentuk menjadi *link list*. Kemudian tahapan berikutnya adalah penemuan *control-flow pattern* berupa relasi XOR, AND, atau OR, penemuan *control-flow pattern* sangat tergantung dari *trace* yang ada dari *event log*. Langkah selanjutnya adalah penemuan *non-free-choice* [11] atau penemuan *invisible task* [12]. Kelebihan pemodelan proses bisnis menggunakan Neo4J adalah fleksibilitas pemodelan yang dapat disesuaikan dengan kebutuhan pengguna atau peneliti. Perangkat lunak seperti ProM hanya dapat melakukan penemuan proses bisnis setelah algoritma yang telah dimasukkan terlebih dahulu, sedangkan dengan menggunakan Neo4J, algoritma dapat dibuat dan dites terlebih dahulu menggunakan satu perangkat lunak sebelum akhirnya algoritma tersebut diimplementasikan.

2.8 *Fitness dan Precision*

Fitness dan precision merupakan salah satu aspek pengukuran untuk mengukur kualitas model proses bisnis [13]. Fitness mengukur berapa banyak *case* yang dapat digambarkan dalam model, sementara precision mengukur berapa banyak *trace* di *event log* dapat digambarkan dalam model. Fitness dan precision bernilai terburuk sebesar 0 hingga bernilai terbaik sebesar 1.

Pada rumus fitness, variabel $n(\text{Captured_Cases})$ akan menyimpan jumlah *case* yang terdapat pada model proses bisnis, sedangkan variabel $n(\text{Cases_in_Event_Log})$ akan menyimpan jumlah *case* yang terdapat pada *event log*. Pada rumus precision, variabel $n(\text{Captured_Traces})$ akan menyimpan jumlah *trace* yang terdapat pada model proses bisnis, sedangkan variabel $n(\text{Traces_in_Event_Log})$ akan menyimpan jumlah *trace* yang terdapat pada *event log*.

$$\text{Fitness}(x) = \frac{n(\text{Captured_Cases})}{n(\text{Cases_in_Event_Log})}$$

$$\text{Precision} = \frac{n(\text{Captured_Traces})}{n(\text{Traces_in_Event_Log})}$$

2.9 *Trace Clustering*

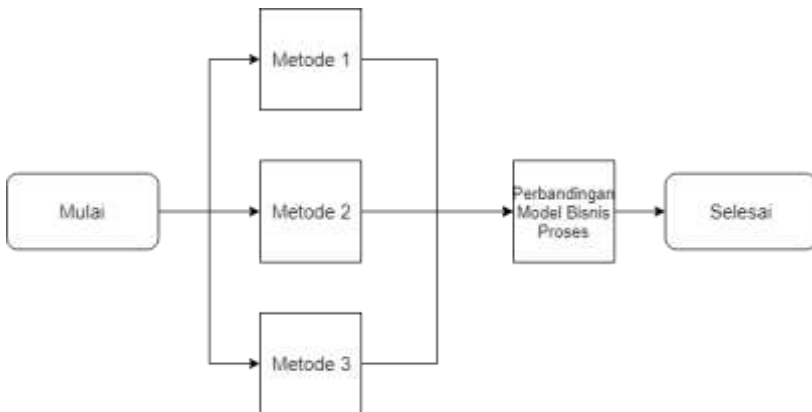
Trace clustering adalah pengelompokan *case* berdasarkan *trace* [14]. Pada [14] *trace clustering* dilakukan dengan membagi set *case* ke dalam kelompok-kelompok yang memiliki list aktivitas atau *trace* yang sama, *trace clustering* juga dapat digunakan untuk mencari *similarity* pada setiap *case*. *Similarity* adalah kemiripan antara model proses bisnis satu dengan model bisnis proses lainnya. *Trace clustering* juga memiliki *fitness* dan *precision* yang tinggi, yaitu dengan nilai sebesar 1 di setiap *traceny*. Contoh *trace clustering*: diketahui *trace* pada suatu *event log* adalah [A, B, C], [A, B, C, D, E], [A, B, C], dan [A, C, D], maka *case* 1 dan 3, memiliki *trace* yang sama dan dianggap sebagai satu kelompok.

[Halaman ini sengaja dikosongkan]

BAB III METODE PEMECAHAN MASALAH

Pada bab ini akan dibahas mengenai metodologi pemecahan masalah yang digunakan sebagai dasar solusi dari pembuatan Tugas Akhir. Metodologi tersebut menerangkan langkah demi langkah proses hingga dapat menghasilkan model proses dari suatu *event log*.

3.1 Cakupan Permasalahan



Gambar 3.1 Alur Pengerjaan Tugas Akhir

Permasalahan utama yang diangkat pada pembuatan Tugas Akhir ini adalah menemukan model proses yang tepat dari *event log* yang memiliki *invisible task* yang merupakan anomali, baik anomali tersebut adalah melewati (*skip*), mengulang (*redo*), dan berpindah (*switch*), dan *non-free choice*. Anomali-anomali tersebut sudah dijelaskan di bab 2.5 dan *non-free-choice* sudah dijelaskan di bab 2.6.

Permasalahan kedua yang dipecahkan pada Tugas Akhir ini adalah membandingkan tiga model proses bisnis yang diimplementasikan dengan menggunakan graf, ketiga metode ini akan dijelaskan pada subbab-subbab selanjutnya. Gambar 3.1 adalah gambaran umum alur proses pengerjaan Tugas Akhir. Setiap metode akan dijelaskan pada subbab-subbab selanjutnya.

3.2 Data

Dalam penelitian ini data yang digunakan adalah *event log* mengenai proses import Request material buku antara perpustakaan dan ruang baca di ITS. *Event log* yang digunakan adalah log sintetis (buatan) yang dibuat sesuai dengan kemungkinan alur yang terjadi saat ILL dan berasal dari beberapa SOP ILL yang telah ada. *Event log* yang digunakan mengandung Case_ID (nomor proses), nama aktivitas, waktu mulai pelaksanaan aktivitas, waktu selesai pelaksanaan aktivitas, dan *actor* (pelaku aktivitas). Data awal berbentuk excel dengan format xls, yang kemudian di konversikan menjadi format csv. Data yang telah berbentuk csv baru bisa di *load* atau dimasukkan ke dalam Neo4J. Contoh *event log* dapat dilihat pada Tabel 3.1.

Event log dibagi menjadi 3, yaitu: *event log* pertama adalah *event log* yang telah dibuat dengan penjelasan seperti paragraf diatas, *event log* kedua adalah *event log* pertama yang telah ditambahkan *invisible task*, dan *event log* ketiga adalah *event log* pertama yang *casenya* diduplikasi.

Tabel 3.1 Contoh Event Log Interlibrary Loan

CaseId	Activity	Start_Timestamp	End_Timestamp
PP1	Request material	02-05-17 07:05	02-05-17 07:10
PP1	Check availability	02-05-17 07:10	02-05-17 10:15
PP1	Confirm material availability	02-05-17 10:15	02-05-17 10:20
PP1	Send material	02-05-17 10:20	02-05-17 14:15
PP1	Confirm material arrival	02-05-17 14:15	02-05-17 14:20
PP1	Take material	02-05-17 14:20	03-05-17 14:35
PP1	Use material	03-05-17 14:35	15-05-17 07:15
PP1	Return material before due time	15-05-17 07:15	15-05-17 07:30
PP1	Send back material	15-05-17 07:30	15-05-17 10:07
PP1	Confirm material send back arrival	15-05-17 10:07	15-05-17 10:12
PP1	Check material condition	15-05-17 10:12	15-05-17 13:12
PP1	Confirm material in good condition	15-05-17 13:12	15-05-17 13:17
PP1	Confirm loan finish	15-05-17 13:17	15-05-17 13:25

3.3 Pembentukan Business Process Model Menggunakan Neo4J

Untuk mendapatkan sebuah model dari Neo4J, hal pertama yang dilakukan yaitu menginputkan *event log* yang berekstensi csv ke

dalam Neo4J dengan cara memasukkan *event log* ke dalam folder import Neo4J, kemudian gunakan kueri import pada Neo4J untuk mengimport *event log*, dan dibuat ke dalam *link list*. Cara untuk menginputkan *event log* yang berekstensi csv ke dalam Neo4J dengan query seperti pada Gambar 3.2 dan menyambungkan relasi dengan menggunakan algoritma penemuan *sequence event log* pada Tabel 3.2. Setelah *link list* terbentuk, selanjutnya adalah melakukan penemuan relasi *control-flow pattern*, yaitu relasi XOR, AND, atau OR. *Event log* akan diimpor dua kali dan dimasukkan kedalam label yang berbeda. Label Activity akan digunakan sebagai label yang menyimpan keseluruhan properti dari *event log*, sedangkan label CaseActivity akan digunakan sebagai label untuk penemuan proses bisnis.

```
LOAD CSV with headers FROM "file:///*.csv"
  AS line
Merge (:Activity {CaseId:line.Case_ID,
Name:line.Activity,
StartTime:line.Start_Timestamp,
EndTime:line.End_Timestamp,
Actor:line.Actor })
LOAD CSV with headers FROM "file:///*.csv"
  AS line
Merge (:CaseActivity {Name:line.Activity })
```

Gambar 3.2 Import Event Log Menggunakan Cypher Query Language

Cara untuk mencari relasi *control-flow pattern* ialah dengan menghitung jumlah *incoming relation* dan *outgoing relation* dari aktivitas. Jumlah *Incoming relation* adalah jumlah aktivitas yang terjadi sebelumnya, sebaliknya dengan *outgoing relation* adalah jumlah aktivitas yang sebelumnya. Setiap relasi control-flow pattern dapat dibagi dua, yaitu relasi *split* dan relasi *join*. Relasi *split* adalah relasi membuka cabang saat aktivitas yang dapat dipilih lebih dari 1, sedangkan relasi *join* adalah relasi yang terjadi saat menutup cabang dari *split*. Algoritma penemuan control-flow pattern juga dapat dilihat pada Tabel 3.3.

Setelah melakukan penemuan *control-flow pattern*, selanjutnya adalah penemuan *invisible task* atau *non-free-choice*. Penemuan *invisible task* atau *non-free-choice* sebenarnya merupakan langkah yang tidak bersangkutan, sehingga penggunaan sangat tergantung oleh kebutuhan pemodelan. Seperti yang telah dijelaskan pada subbab 2.5, *invisible task* dibagi menjadi tiga, yaitu Melewati (Skip), Berpindah (Switch), dan Mengulang (Redo). Algoritma *invisible task* dengan tipe Melewati dicantumkan pada Tabel 3.2, sedangkan algoritma *invisible task* dengan tipe Berpindah dan Mengulang dicantumkan pada Tabel 3.3. Penemuan *non-free-choice* dicantumkan pada Tabel 3.3.

Algoritma penemuan *Sequence event log*, *control-flow pattern* XOR dan *invisible task* pada Tabel 3.2 merupakan hasil penelitian sebelumnya pada [11] dan [12]. Penemuan *non-free-choice* yang sebelumnya telah diteliti pada [11] telah dimodifikasi pembuatan relasinya dari menggunakan fungsi *create* menjadi menggunakan fungsi *merge*, sedangkan penemuan *control-flow pattern* AND dan OR merupakan penemuan baru.

Contoh hasil penemuan pattern sequence dan control-flow ada pada Gambar 2.1. Contoh hasil penemuan *invisible task* ada pada Gambar 2.5, sedangkan contoh hasil penemuan *non-free-choice* ada pada Gambar 2.7.

Tabel 3.2 Algoritma Penemuan dalam Neo4J

Pattern		Cypher
Sequence Event Log		Algoritma yang digunakan pada pattern ini dicantumkan pada Tabel 3.3
Control-Flow Pattern	XOR Split	
	XOR Join	
	AND Split	<pre>match(activity[i])-[:NEXT]->(activity[i+1]) if outgoing(activity[i])>1 and outgoing(activity[i+1])>=2 create relation activity[i]-[:ANDSPLIT]-> activity [i+1] delete NEXT</pre>
	AND Join	<pre>match(activity[i])-[:NEXT]->(activity[i+1])</pre>

Pattern		Cypher
		<pre>if outgoing(activity[i])>=2 and incoming(activity[i+1])=2 create relation activity[i]-[r:XORJOIN]-> activity [i+1] delete NEXT</pre>
	OR Split	<pre>match(activity[i])-[r:NEXT]->(activity[i+1]) if outgoing(activity[i])>1 and outgoing(activity[i+1])>=2 create relation activity[i]-[r:ORSPLIT]-> activity [i+1] delete NEXT</pre>
	OR Join	<pre>match(activity[i])-[r:NEXT]->(activity[i+1]) if outgoing(activity[i])>=2 and incoming(activity[i+1])=3 create relation activity[i]-[r:XORJOIN]-> activity [i+1] delete NEXT</pre>
Invisible task	Skip	<pre>match(activity[i])-[r:NEXT]->(activity[i+1]) where outgoing(activity[i])>1 and ingoing(activity[i+1])>1 and outgoing(activity[i+1])>1 create node i:invisible task create relation activity[i]-(NEXT)->i create relation i-(NEXT)->activity[i+1] delete relation activity[i]-(NEXT)->activity[i+1]</pre>
	Switch	Algoritma yang digunakan pada pattern ini dicantumkan pada Tabel 3.3
	Redo	
Non-free-choice		

3.4 Metode Pertama



Gambar 3.3 Alur Pengerjaan Metode Pertama

Masukan berupa *event log* yang telah dijelaskan di subbab 3.2. *Event log* berisi 13 *case* dengan masing-masing *case* memiliki *trace* yang berbeda, aktivitas sebanyak 220 buah dengan waktu mulai dan waktu selesai yang berbeda untuk setiap aktivitas. Keseluruhan *event log* yang digunakan pada metode pertama ada pada lampiran Tabel 7.4.

Metode pertama adalah membentuk graf dengan penemuan proses bisnis yang umum dilakukan, yaitu dengan cara membaca atau melakukan penemuan pada keseluruhan *event log*. Gambar 3.3

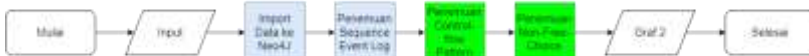
merupakan gambaran alur pada metode pertama. Metode pertama memodelkan proses bisnis dalam bentuk graf yang mencakup *load* atau *import data event log* ke dalam Neo4J, penemuan sequence, penemuan *control-flow pattern*, penemuan *invisible task*, dan penemuan *non-free-choice*. Hasil dari penemuan *invisible task* dari model pertama akan dijadikan acuan penambahan *invisible task* pada *event log* yang akan digunakan pada metode kedua.

Tabel 3.3 Algoritma yang digunakan pada Metode Pertama

Langkah	Pattern		Algoritma
2	Sequence Event Log		<i>match (activity[i])</i> <i>for i=0 until i=count_activity</i> <i>match (activity[I]). (activity[I+1])</i> <i>if (activity[i]).CaseId=</i> <i>(activity[i+1]).CaseId and</i> <i>(activity[i]).Name=(activity[I]).Name and</i> <i>(activity[i+1]).Name=</i> <i>(activity[I+1]).Name</i> <i>create relation activity[I]-[r:NEXT]-</i> <i>>activity[I+1]</i>
3	XOR	Split	<i>match(activity[i])-[r:NEXT]-</i> <i>>(activity[i+1])</i> <i>if outgoing(activity[i])>1 and</i> <i>incoming(activity[i+1])=1</i> <i>create relation activity[i]-[r:XORSPLIT]-</i> <i>> activity [i+1]</i> <i>delete NEXT</i>
		Join	<i>match(activity[i])-[r:NEXT]-</i> <i>>(activity[i+1])</i> <i>if outgoing(activity[i])=1 and</i> <i>incoming(activity[i+1])>1</i> <i>create relation activity[i]-[r:XORJOIN]-></i> <i>activity [i+1]</i> <i>delete NEXT</i>
4	Invisible Task	Switch	<i>match(activity[i])-[r:NEXT]-</i> <i>>(activity[i+1])</i> <i>where outgoing(activity[i])>1 and</i> <i>ingoing(activity[i+1])>1</i> <i>create node i:invisible task</i> <i>create relation activity[i]-(- NEXT)->i</i> <i>create relation i-(- NEXT)->activity[i+1]</i>

Langkah	Pattern	Algoritma
		<i>delete relation activity[i]-[:NEXT]->activity[i+1]</i>
	<i>Redo</i>	<i>match(activity[i])-[:r:NEXT]->(activity[i+1])</i> <i>create relation activity[i]-[:r:NEXT]->activity[i+1]</i>
5	<i>Non-free-choice</i>	<i>match(activity[i])-[:XORJOIN]->(activity[i+1])-[:XORSPLIT]->(activity[i+2])</i> <i>match(activity[I])-[:SEQUENCE]->(activity[I+1])-[:SEQUENCE]->(activity[I+2])</i> <i>where activity[i].name = activity[I].name</i> <i>and activity[I+1].name = activity[I+1].name and</i> <i>activity[I+2].name = activity[I+2].name</i> <i>merge relation activity[i]-[:NONFREECHOICE]->activity[i+2]</i>

3.5 Metode Kedua



Gambar 3.4 Alur Pengerjaan Metode Kedua

Masukan berupa *event log* yang digunakan pada metode pertama yang telah ditambahkan *invisible task*. Jumlah *case* sebanyak 13 *case* dengan jumlah aktivitas termasuk *invisible task* sebanyak 342. *Invisible task* sejumlah 122 dengan varian sebanyak 24 jenis. Sama dengan *event log* pada metode pertama, setiap aktivitas memiliki waktu mulai dan waktu selesai masing-masing. Aktivitas *invisible task* memiliki durasi selama satu menit dan waktu diambil dari aktivitas sebelumnya yang dikurangi waktunya sebanyak satu menit. *Event log interlibrary loan* hanya berisi *invisible task* dengan tipe berpindah dan mengulang. Keseluruhan *event log* yang digunakan pada metode kedua ada pada lampiran Tabel 7.5.

Metode kedua adalah pembentukan graf dengan *event log* yang mengandung *invisible task* dan penemuan sequence dilakukan dengan metode yang sama seperti metode pertama. Gambar 3.4

merupakan gambaran alur pengerjaan metode kedua, dapat dilihat bahwa proses pengerjaan pada metode kedua berbeda dengan proses pengerjaan pada metode pertama. Perbedaan metode pertama dan kedua adalah pada proses penemuan *control-flow pattern* dan *non-free-choice*. Perbedaan penemuan *control-flow pattern* dan *non-free-choice* akan dicantumkan pada Tabel 3.4.

Tabel 3.4 Algoritma yang digunakan pada Metode Kedua

Langkah	Pattern		Algoritma
2	Sequence Event Log		Sama dengan langkah 2 pada Tabel 3.3.
3	XOR	Split	<i>match(activity[i])-[:NEXT]->(activity[i+1])->[:NEXT]->(activity[i+2])</i> <i>if outgoing(activity[i])>1 and incoming(activity[i+1])=1 and activity[i+1].Name != Invisible task</i> <i>create relation activity[i]-[r:XORSPLIT]-> activity [i+2]</i> <i>delete NEXT</i>
		Join	<i>match(activity[i])-[:NEXT]->(activity[i+1])->[:NEXT]->(activity[i+2])</i> <i>if outgoing(activity[i])=1 and incoming(activity[i+1])>1 and activity[i+1].Name != Invisible task</i> <i>create relation activity[i]-[r:XORJOIN]-> activity [i+2]</i> <i>delete NEXT</i>
4	Non-free-choice		<i>match (activity[i])-[:NEXT]->(activity[i+1])-[:NEXT]->(activity[I+2])-[:NEXT]->(activity[i+3])-[:NEXT]->(activity[i+4])</i> <i>match (activity[I])-[:SEQUENCE]->(activity[I+1])-[:SEQUENCE]->(activity[I+2])-[:SEQUENCE]->(activity[I+3])-[:SEQUENCE]->(activity[I+4])</i> <i>match (activity[i])-[:XORJOIN]->(activity[i+2])-[:XORSPLIT]->(activity[i+4])</i> <i>where activity[I].Name= activity[i].Name and activity[I+2]= activity[i+2].Name and (activity[I+4]).Name=(activity[i+4]).Name</i> <i>merge ()-[:NONFREECHOICE]->(e)</i>

Case_ID	Activity
PP1	A
PP1	B
PP1	D
PP2	A
PP2	C
PP2	D
PP3	A
PP3	B
PP3	D

Case_ID	Activity
PP1	A
PP1	Invisible task 1
PP1	B
PP1	Invisible task 2
PP1	D
PP2	A
PP2	Invisible task 3
PP2	C
PP2	Invisible task 4
PP2	D
PP3	A
PP3	Invisible task 1
PP3	B
PP3	Invisible task 2
PP3	D

Gambar 3.5 Contoh Penambahan *Invisible Task* pada *Event Log* dengan Tipe Berpindah

Cara penambahan *invisible task* ke dalam *event log* dapat dibedakan sesuai dengan tipe *invisible task*. Berikut adalah peraturan untuk menambahkan *invisible task* dengan tipe berpindah ke dalam *event log*:

1. Penambahan *invisible task* di dalam *event log* diurutkan berdasarkan *case* dan berdasarkan urutan percabangan.
2. *Invisible task* akan selalu muncul ketika terjadi *split* atau *join*.

Contoh: pada Gambar 3.5 case_ID PP1 dan PP2, *invisible task* akan muncul diantara aktivitas A dan B pada case_ID PP1 dan aktivitas A dan C pada case_ID PP2.

3. *Invisible task* harus diberi penomoran yang berbeda diantara 2 aktivitas yang berbeda.

Contoh: pada Gambar 3.5 case_ID PP1, *invisible task* diantara aktivitas A dan B harus memiliki penomoran yang berbeda dengan *invisible task* diantara aktivitas B dan D.

4. *Invisible task* yang akan muncul diantara dua aktivitas yang sama pada *case* yang berbeda harus diberi penomoran yang sama.

Contoh: pada Gambar 3.5 case_ID PP1 dan PP3, *invisible task* diantara aktivitas A dan B pada case_ID PP1 harus memiliki penomoran yang sama dengan *invisible task* diantara aktivitas A dan B pada case_ID PP3.

Case_ID	Activity	Case_ID	Activity
PP1	A	PP1	A
PP1	B	PP1	B
PP1	C	PP1	C
PP1	D	PP1	D
PP1	E	PP1	E
PP2	A	PP2	A
PP2	B	PP2	B
PP2	C	PP2	C
PP2	B	PP2	Invisible task 1
PP2	C	PP2	B
PP2	B	PP2	C
PP2	C	PP2	Invisible task 1
PP2	D	PP2	B
PP2	E	PP2	C
PP2	D	PP2	D
PP2	E	PP2	E
		PP2	Invisible task 2
		PP2	D
		PP2	E

Gambar 3.6 Contoh Penambahan *Invisible Task* pada *Event Log* dengan Tipe Mengulang

Berikut adalah peraturan untuk menambahkan *invisible task* dengan tipe mengulang ke dalam *event log*:

1. Penambahan *invisible task* berdasarkan apabila ada aktivitas yang diulang.

Contoh: pada Gambar 3.6 case_ID PP1 tidak memiliki aktivitas yang diulang, sehingga *invisible task* tidak perlu ditambahkan pada case_ID PP1. Sebaliknya, case_ID PP2 memiliki aktivitas yang diulang, sehingga *invisible task* perlu ditambahkan pada case_ID PP2.

2. Penambahan *invisible task* sesuai dengan jumlah perulangan aktivitas yang ada pada suatu *case*.

Contoh: pada Gambar 3.6 case_ID PP2, ada beberapa aktivitas yang diulang, maka penambahan *invisible task* harus sebanyak jumlah pengulangan aktivitas.

3. Penambahan *invisible task* diantara dua aktivitas yang berbeda harus memiliki penomoran yang berbeda.

Contoh: pada Gambar 3.6 case_ID PP2, *invisible task* diantara aktivitas C dan B harus memiliki penomoran yang berbeda dengan *invisible task* diantara aktivitas E dengan D.

4. Penambahan *invisible task* diantara dua aktivitas yang sama di dalam *case* yang berbeda atau *case* yang sama harus diberi penomoran yang sama.

Contoh: pada Gambar 3.6 case_ID PP2, *invisible task* diantara aktivitas C dan B pada perulangan pertama harus memiliki penomoran yang sama dengan *invisible task* diantara aktivitas C dan B pada perulangan kedua.

Case_ID	Activity	Case_ID	Activity
PP1	A	PP1	A
PP1	B	PP1	B
PP1	C	PP1	C
PP1	D	PP1	D
PP1	F	PP1	F
PP2	A	PP2	A
PP2	C	PP2	invisible task 1
PP2	E	PP2	C
PP2		PP2	invisible task 2
PP3	A	PP2	E
PP3	C	PP3	A
PP3	D	PP3	invisible task 1
PP3	E	PP3	C
		PP3	D
		PP3	E

Gambar 3.7 Contoh Penambahan *Invisible Task* pada *Event Log* dengan Tipe Melewati

Berikut adalah peraturan untuk menambahkan *invisible task* dengan tipe melewati ke dalam *event log*:

1. Penambahan *invisible task* berdasarkan ada atau tidaknya aktivitas yang dilewati dalam suatu *event log*.

Contoh: pada Gambar 3.7 case_ID PP1 tidak memiliki aktivitas yang dilewati, sehingga *invisible task* tidak perlu ditambahkan. Sebaliknya pada case_ID PP2 ada aktivitas yang dilewati, sehingga *invisible task* perlu ditambahkan

2. Penambahan *invisible task* diantara dua aktivitas yang berbeda diberi nama atau penomoran yang berbeda.

Contoh: pada Gambar 3.7 case_ID PP2, *invisible task* diantara aktivitas A dan C harus memiliki penomoran yang berbeda dengan *invisible task* diantara aktivitas C dan E.

3. Penambahan *invisible task* diantara dua aktivitas yang sama pada *case* yang berbeda diberi penomoran yang sama.

Contoh: pada Gambar 3.7 case_ID PP2 dan PP3, *invisible task* diantara aktivitas A dan C pada case_ID PP2 harus memiliki penomoran yang sama dengan *invisible task* diantara aktivitas A dan C pada case_ID PP3.

3.6 Metode Ketiga



Gambar 3.8 Alur Pengerjaan Metode Ketiga

Masukan berupa *event log* pada metode pertama yang telah di duplikasi datanya. Jumlah *case* sebanyak 70 dengan jumlah aktivitas sebanyak 1198. Berbeda dengan *event log* pada metode pertama dan kedua, *event log* pada metode ketiga tidak memiliki waktu mulai, waktu selesai, dan aktor.

Metode ketiga adalah membentuk graf yang dari data yang *tracenya* telah di saring dan menggunakan *event log* yang *casenya* telah di duplikasi untuk melihat apakah *trace clustering* dapat berjalan dengan benar.

Langkah-langkah pengerjaan metode ketiga lebih rumit dibandingkan pengerjaan metode pertama maupun kedua, Gambar 3.8 merupakan gambaran proses pengerjaan metode ketiga. *Event log* yang telah dibuat diduplikasi casenya, kemudian hasilnya digunakan untuk pemodelan metode ketiga. Setelah melakukan *load* atau *import* data *event log* ke Neo4J langkah selanjutnya adalah penemuan *sequence* setiap *case*, hal ini dilakukan untuk memisahkan setiap *case* yang ada pada *event log*. Kemudian setelah penemuan *sequence*, *trace clustering* dan penyaringan data yang mengandung anomali baru dapat dilakukan. Setelah data yang mengandung anomali dihapus, langkah berikutnya adalah menghapus relasi atau menggabungkan node yang memiliki nama aktivitas sama. Kedua langkah ini bisa dilakukan tidak berurutan. Setelah kedua langkah ini dilakukan, perlu dilakukan penghapusan *caseId* pada label kedua, pada contoh kali ini

adalah label CaseActivity. Kemudian langkah selanjutnya sama dengan langkah-langkah yang dilakukan pada metode pertama.

Tabel 3.5 Algoritma yang digunakan pada Metode Ketiga

Langkah	Pattern	Algoritma
2	<i>Sequence Case</i>	<code>match (activity[i])</code> <code>for i=0 until i=count_activity</code> <code>match (activity[I]). (activity[I+1])</code> <code>if (activity[i]).CaseId= (activity[i+1]).CaseId and</code> <code>(activity[i]).Name=(activity[I]).Name and</code> <code>(activity[i+1]).Name=(activity[I+1]).Name and</code> <code>(activity[i]).CaseId =(activity[I]).CaseId and</code> <code>(activity[i+1]).CaseId =(activity[I+1]).CaseId</code> <code>create relation activity[I]-[r:NEXT]->activity[I+1]</code>
6	<i>Sequence Event Log</i>	Sama dengan langkah 2 pada Tabel 3.3
7	<i>Control-Flow Pattern</i>	Sama dengan langkah 3 pada Tabel 3.3
8	<i>Invisible Task</i>	Sama dengan langkah 4 pada Tabel 3.3
9	<i>Non-free-choice</i>	Sama dengan langkah 5 pada Tabel 3.3

3.7 Perbandingan Ketiga Metode

Langkah terakhir adalah membandingkan graf yang telah terbentuk dari ketiga tahapan yang telah dilakukan sebelumnya untuk melihat graf mana yang memberikan tingkat kebenaran yang paling tinggi. Ketiga tahapan tadi nantinya juga akan digambarkan menggunakan ProM. ProM digunakan sebagai pembanding apakah relasi yang terbentuk pada ProM sudah sesuai dengan graf yang terbentuk pada Neo4J. Setiap model graf yang terbentuk juga akan dibandingkan *fitness* dan *precisionnya*. Kemudian hasil dari perbandingan akan dituliskan pada bab 5 untuk pembahasan dan 6 sebagai kesimpulan dari pengerjaan Tugas Akhir.

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Bab ini membahas tentang kueri Cypher yang akan digunakan untuk pemodelan proses bisnis menggunakan Neo4J.

4.1 Lingkungan Implementasi

Lingkungan implementasi merupakan lingkungan dimana sistem ini dibangun, dimana akan dijelaskan mengenai kebutuhan perangkat yang diperlukan untuk membangun sistem ini. Lingkungan implementasi dibagi menjadi dua, yaitu lingkungan implementasi terhadap perangkat keras dan lingkungan implementasi terhadap perangkat lunak.

4.1.1. Perangkat Keras

Implementasi dilakukan pada sebuah laptop dengan spesifikasi sebagai berikut:

- Merk : ACER.
- Seri : Aspire E14 E5-476G.
- Prosesor : Intel Core i5-8250U @ 1.60 GHz.
- RAM : 8 GB.

4.1.2. Perangkat Lunak

Perangkat lunak yang mendukung pengerjaan Tugas Akhir adalah:

1. Sistem Operasi Windows
Sistem operasi yang digunakan adalah Microsoft Windows 10 Professional 64-bit.
2. ProM 5.2.
3. Neo4j Dekstop 1.0.23
4. Microsoft Excel.

4.2 Penjelasan Implementasi

Pada sub bab ini dijelaskan implementasi setiap metode yang dijelaskan pada bab metode pemecahan masalah, sehingga terbentuk suatu perangkat lunak yang mengimplementasi metode-metode pada bab metode pemecahan masalah.

4.2.1. Implementasi Metode Pertama

4.2.1.1. Pembentukan Business Process Model dengan Neo4J

Untuk mengimplementasikan pembentukan *Business Process model* dengan Neo4J, langkah pertama kita harus membuat *link list* dari *event log* yang berbentuk csv, kueri langkah ini ada di Kueri Cypher 4.1.

```
LOAD CSV with headers FROM "file:///*.csv"
  AS line
Merge (:Activity {CaseId:line.Case_ID,
Name:line.Activity,
StartTime:line.Start_Timestamp,
EndTime:line.End_Timestamp,
Actor:line.Actor })
LOAD CSV with headers FROM "file:///*.csv"
  AS line
Merge (:CaseActivity {Name:line.Activity })
//Discovery Sequence
match (c:Activity)
with COLLECT(c) AS Caselist
unwind range(0,Size(Caselist) - 2) as idx
with Caselist[idx] AS s1, Caselist[idx+1] AS s2
match (b:CaseActivity),(a:CaseActivity)
where s1.CaseId = s2.CaseId AND s1.Name = a.Name AND s2.Name = b.Name
merge (a)-[:NEXT {relation:"NEXT"}]->(b)
```

Kueri Cypher 4.1 Mengubah Event Log ke Link List

4.2.1.1 Implementasi Penemuan Control-Flow Pattern

Setelah berhasil mendapatkan *link list*, maka perlu dilakukan deteksi relasi antar aktivitas dengan melihat *outgoing relation* dan *incoming relation* seperti pada Kueri Cypher 4.2. Sedangkan relasi AND dan OR sedikit lebih berbeda dengan relasi XOR. Relasi XOR selalu menghitung *incoming* dan *outgoing relation* baik itu ketika *split* maupun *join*, sedangkan AND dan OR hanya menghitung *outgoing*

relation pada saat *split* dan penghitungan *incoming* dan *outgoing relation* baru dilakukan pada saat *join*. Berhubung tidak adanya relasi AND dan OR maka hanya relasi XOR yang akan digunakan pada pemodelan ini. Untuk kueri penemuan relasi AND ada pada Kueri Cypher 4.3, sedangkan untuk kueri penemuan relasi OR ada pada Kueri Cypher 4.4.

```
match (n)-[:NEXT]->(a)
where size((n)-->()) > 1 and (size((a)<--()) = 1)
create (n)-[:XORSPLIT {relation:"XOR Split"}]->(a)
return distinct 'XORSPLIT', n.Activity, a.Activity
match (n)-[:NEXT]->(a)
where size((n)-->()) = 1 and (size((a)<--()) > 1 )
create (n)-[:XORJOIN {relation:"XOR Join"}]->(a)
return distinct 'XORJOIN', n.Activity, a.Activity
```

Kueri Cypher 4.2 Penemuan XOR di Neo4J

```
match (n)-[:NEXT]->(a)
where size((n)-->()) > 1 and size((a)-->()) >= 2
create (n)-[:ANDSPLIT {relation:"AND Split"}]->(a)
return distinct 'ANDSPLIT', n.Activity, a.Activity
match (n)-[:NEXT]->(a)
where size((n)-->()) >= 2 and size((a)<--()) = 2
create (n)-[:ANDJOIN {relation:"AND Join"}]->(a)
return distinct 'ANDJOIN', n.Activity, a.Activity
```

Kueri Cypher 4.3 Penemuan AND di Neo4J

```
match (n)-[:NEXT]->(a)
where size((n)-->()) > 1 and size((a)-->()) >= 2
create (n)-[:ORSPLIT {relation:"OR Split"}]->(a)
return distinct 'ORSPLIT', n.Activity, a.Activity
match (n)-[:NEXT]->(a)
where size((n)-->()) >= 2 and size((a)<--()) = 3
create (n)-[:ORJOIN {relation:"OR Join"}]->(a)
return distinct 'ORJOIN', n.Activity, a.Activity
```

Kueri Cypher 4.4 Penemuan OR di Neo4J

4.2.1.2 Penemuan Invisible Task dan Non-Free-Choice

Setelah mengetahui relasi *control-flow pattern*, selanjutnya kita perlu melakukan penemuan *invisible task* dan *non-free-choice* dari model graf yang telah terbentuk sebelumnya, menggunakan Kueri Cypher 4.5 dan Kueri Cypher 4.6.

```
match (n)-[r:NEXT]->(a)
where size ((n)-->())>1 and size ((a)<-->())>1
create (i:invisibletask {name:"invisible task"})
create (n)-[:NEXT]->(i)
create (i)-[:NEXT]->(a)
delete r
```

Kueri Cypher 4.5 Penemuan *Invisible Task*

```
LOAD CSV WITH HEADERS FROM "file:///*.csv" AS line
Merge (n:SeqActivity {
  CaseId:line.Case_ID,
  Name:line.Activity,
  StartTime:line.Start_Timestamp,
  EndTime:line.End_Timestamp
})
With n Order by n.Start_Timestamp ASC
With COLLECT(n) AS activities
Foreach (m IN RANGE(0, SIZE(activities)-2) |
  Foreach (prec IN [activities[m]] |
    Foreach (next IN [activities[m+1]] |
      Merge (prec)-[:SEQUENCE]->(next))))

match p=(a)-[r:XORJOIN]->(n)-[s:XORSPLIT]->(b)
match q=(x)-[:SEQUENCE]->(y)-[:SEQUENCE]->(z)
where x.Name=a.Name and y.Name=n.Name and z.Name=b.Name
merge (a)-[:NONFREECHOICE]->(b)
delete r,s
```

Kueri Cypher 4.6 Pembentukan Model Proses Bisnis *Sequence* dan Penemuan *Non-Free-Choice*

4.2.2 Implementasi Metode Kedua

4.2.2.1 Pembentukan Process Model dengan Neo4J

Pada metode kedua load data dan penemuan sequence menggunakan cara yang sama dengan metode pertama, sehingga kueri yang digunakan adalah Kueri Cypher 4.1.

4.2.2.2 Penemuan *Control-Flow-Pattern* pada *Event Log* yang mengandung *Invisible Task*

Berbeda dengan metode pertama, penemuan *control-flow pattern* pada *event log* yang mengandung *invisible task* sedikit lebih rumit. Pada metode pertama, *control-flow pattern* menghubungkan *node* bercabang dengan *node* selanjutnya, sedangkan pada metode kedua *control-flow pattern* harus menghubungkan *node* yang tidak memiliki nama aktivitas *Invisible task*. Berhubung event log hanya mengandung relasi *control-flow pattern* berupa XOR, Kueri Cypher 4.7 merupakan penemuan *control-flow pattern* XOR pada *event log* yang mengandung *invisible task*.

```

match (n)-[:NEXT]->(a)-[:NEXT]->(b)
where size{(n)-->{}} > 1 and (size{(a)-->{}} = 1) and a.Name <> "Invisible task "
create (n)-[:XORSPLIT {relation:"XOR Split"}]->(b)
return distinct 'XORSPLIT', n.Activity, a.Activity

match (n)-[:NEXT]->(a)-[:NEXT]->(b)
where size{(a)-->{}} = 1 and (size{(b)-->{}} > 1 ) and a.Name <> "Invisible task "
create (n)-[:XORJOIN {relation:"XOR Join"}]->(b)
return distinct 'XORJOIN', n.Activity, a.Activity

match ()-[:XORSPLIT]->{}
match ()-[a:XORJOIN]->{}
detach delete a

```

Kueri Cypher 4.7 Penemuan *Control-Flow Pattern* XOR pada *Event Log* yang Mengandung *Invisible Task*

4.2.2.3 Penemuan *Non-Free-Choice* pada *Event Log* yang Mengandung *Invisible Task*

Sama halnya dengan subbab sebelumnya, penemuan *non-free-choice* pada *event log* yang mengandung *invisible task* pada metode kedua berbeda dengan penemuan *non-free-choice* pada metode pertama karena *non-free-choice* membaca *control-flow*

pattern XORSPLIT dan XORJOIN untuk menentukan apakah suatu *event log* mengandung *non-free-choice* atau tidak. Berhubung pada metode kedua penemuan *control-flow pattern* menggunakan cara yang berbeda dengan metode pertama, maka penemuan *non-free-choice* juga harus diubah, akan tetapi pembentukan model proses bisnis *sequence* tidak perlu diubah, sehingga pembentukan model proses bisnis *sequence* pada Kueri Cypher 4.6 tetap bisa digunakan. Kueri Cypher 4.8 merupakan kueri penemuan *non-free-choice* pada *event log* yang mengandung *invisible task*.

```
match (a)-[:NEXT]->(b)-[:NEXT]->(c)-[:NEXT]->(d)-[:NEXT]->(e)
match (v)-[:SEQUENCE]->(w)-[:SEQUENCE]->(x)-[:SEQUENCE]->(y)-[:SEQUENCE]->(z)
match (a)-[r:XORJOIN]->(c)-[s:XORSPLIT]->(e)
where v.Name=a.Name and x.Name=c.Name and z.Name=e.Name
merge (a)-[:NONFREECHOICE]->(e)
delete r,s
```

Kueri Cypher 4.8 Penemuan Non-Free-Choice pada Event Log yang Mengandung Invisible Task

4.2.3 Implementasi Metode Ketiga

4.2.3.1 Pembentukan Process Model Berdasarkan Case dengan Neo4J

Berbeda dengan *import* atau *load* data pada metode pertama, untuk melakukan *trace clustering*, *event log* yang akan di *import* atau di *load* tidak perlu di gabungkan aktivitasnya, karena dengan melakukan penggabungan aktivitas maka Neo4J tidak akan bisa melakukan perhitungan *trace* dan melihat *trace* apa saja yang ada di *event log*. Kueri Cypher 4.9 merupakan modifikasi dari Kueri Cypher 4.1 dengan beberapa perbedaan yaitu pada saat load data tidak menggunakan fungsi *merge* tapi menggunakan fungsi *create*. Kemudian untuk penemuan *sequence* dipisahkan sesuai CaseId pada *event log* sehingga pada metode ketiga, model proses bisnis yang terbentuk nantinya akan terpisah menjadi potongan-potongan *trace* berbeda dengan metode pertama dengan model proses bisnis yang sudah terbentuk secara langsung dengan penggabungan *node* dan penemuan *sequence* dari *node* yang telah digabungkan sebelumnya.


```

LOAD CSV with headers FROM "file:///*.csv"
  AS line
Create (:Activity {CaseId:line.Case_ID,
Name:line.Activity,
StartTime:line.Start_Timestamp,
EndTime:line.End_Timestamp,
Actor:line.Actor })
LOAD CSV with headers FROM "file:///*.csv"
  AS line
Create (:CaseActivity {CaseId:line.Case_ID,Name:line.Activity })

match (c:Activity)
with collect(c) AS Caselist
unwind range(0,Size(Caselist) - 2) as idx
with Caselist[idx] AS s1, Caselist[idx+1] AS s2
match (b:CaseActivity),(a:CaseActivity)
where s1.CaseId = s2.CaseId AND
s1.Name = a.Name AND
s2.Name = b.Name AND
s1.CaseId = a.CaseId AND
s2.CaseId = b.CaseId
merge (a)-[:NEXT {relation:"NEXT"}]->(b)

```

Kueri Cypher 4.9 Mengubah *Event Log* ke *Link List* dengan dipisahkan Setiap *Case*

4.2.3.2 Trace Clustering dan Penyaringan Data Anomali

Model proses bisnis yang telah terbentuk setelah melakukan *load* data dan penemuan *sequence* kemudian ditampilkan sesuai *trace* dengan menggunakan Kueri Cypher 4.10. Data yang ditampilkan dari Kueri Cypher 4.10 adalah case apa saja yang memiliki trace x, jumlah case yang memiliki trace x, dan trace dari event log itu sendiri.

```

match(a:Activity)
with a.CaseId as id,
collect (a.Name) as Trace_Type
match(b:CaseActivity)
where id = b.CaseId
return count (distinct b.CaseId) as Frequencies,
Trace_Type, collect (distinct b.CaseId) as CaseId
order by Frequencies desc

```

Kueri Cypher 4.10 *Trace Clustering* dan Penghitungan Frekuensi *Case*

Hasil dari implementasi menggunakan Kueri Cypher 4.10 yang memiliki frekuensi paling kecil dianggap sebagai data anomali dan kemudian dihapus menggunakan Kueri Cypher 4.11. Pada Kueri Cypher 4.11, dari kueri pertama akan didapatkan daftar panjang *trace* pada setiap *case*. Kemudian pada kueri kedua *Frequencies* perlu diganti oleh panjang *trace* yang akan dihapus.

```

match (m:CaseActivity)
with m.CaseId as id, collect (m.Name) as Activities, count (m.Name) as Frequencies
return count(id) as ID_Frequencies,
collect(id) as IDs, Activities,
Frequencies as Activities_Count
order by Activities_Count desc
|
match (m:CaseActivity)
with m.CaseId as id, collect (m.Name) as Activities, count (m.Name) as Frequencies
where Frequencies=6
match (n:CaseActivity{CaseId:id})
detach delete n

```

Kueri Cypher 4.11 Penghapusan Data Anomali

4.2.3.3 Penggabungan Node dan Penghapusan Relasi

Setelah data bersih dari anomali, langkah selanjutnya adalah menggabungkan aktivitas yang memiliki nama sama pada setiap *case* dengan menggunakan kueri pada Kueri Cypher 4.12 atau melakukan penghapusan relasi *sequence* menggunakan kueri pada Kueri Cypher 4.14. Kedua langkah ini dapat dilakukan tidak berurutan. Untuk menjalankan Kueri Cypher 4.12 dibutuhkan plugin apoc. Setelah penggabungan node dan penghapusan node dilakukan, *caseId* pada label kedua harus dihapus dengan Kueri Cypher 4.14.

```

match (n:*)
with n.Name AS name, collect(n) AS nodelist, count(*) AS count
where count > 1
call apoc.refactor.mergeNodes(nodelist) YIELD node
return node

```

Kueri Cypher 4.12 Penggabungan Node

```
match ()-[n:NEXT]->()
detach delete n
```

Kueri Cypher 4.13 Penghapusan Relasi NEXT

```
match (n:CaseActivity)
remove n.CaseId
return n
```

Kueri Cypher 4.14 Penghapusan CaseId

4.2.3.4 Penemuan *Sequence Ulang*, *Control-Flow Pattern*, *Invisible Task*, dan *Non-Free-Choice*

Setelah melakukan semua proses pada subbab 4.2.3.3 proses selanjutnya adalah penemuan sequence ulang, karena kondisi graf terpisah dikarenakan relasi sequence sebelumnya telah dihapus. Penemuan sequence ulang menggunakan kueri 4.1 tanpa perlu *load* data ulang. Kemudian sama dengan metode pertama langkah-langkah selanjutnya adalah penemuan *control-flow pattern*, penemuan *invisible task*, dan penemuan *non-free-choice*.

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini membahas hasil dan pembahasan pada aplikasi yang dikembangkan. Pada bab ini akan dijelaskan tentang data yang digunakan, hasil yang didapatkan dari penggunaan perangkat lunak dan uji coba yang dilakukan pada perangkat lunak yang telah dikerjakan untuk menguji apakah kueri pada perangkat lunak telah diimplementasikan dengan benar dan berjalan sebagaimana mestinya.

5.1 Lingkungan Uji Coba

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi pembuatan sistem pada tugas akhir ini. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

1. Perangkat keras
 - a. Prosesor : Intel Core i5-8250U @ 1.60GHz.
 - b. Memori (RAM) : 8 GB.
 - c. Tipe sistem : 64-bit sistem operasi.
2. Perangkat lunak
 - a. Sistem operasi : Windows 10.
 - b. Kakas bantu : Neo4j.

5.2 Keluaran

5.2.1 Metode Pengujian

Pada hasil dari tugas akhir ini terdapat 3 graf yang menggambarkan model proses bisnis yaitu graf yang berasal dari penemuan secara biasa, *graph* yang berasal dari penemuan event log yang berisi *invisible task*, dan graf yang berasal dari hasil penyaringan *trace* data. Untuk menguji baik buruk nya metode graf *database* ini, maka kita mengkomparasi antara 3 graf ini dan mengetahui graf mana yang merupakan paling baik.

5.2.2 Skenario Uji Coba

Skenario uji coba berdasarkan metode implementasi yang dijelaskan pada bab 3. Terdapat beberapa pengujian yang dilakukan, yaitu:

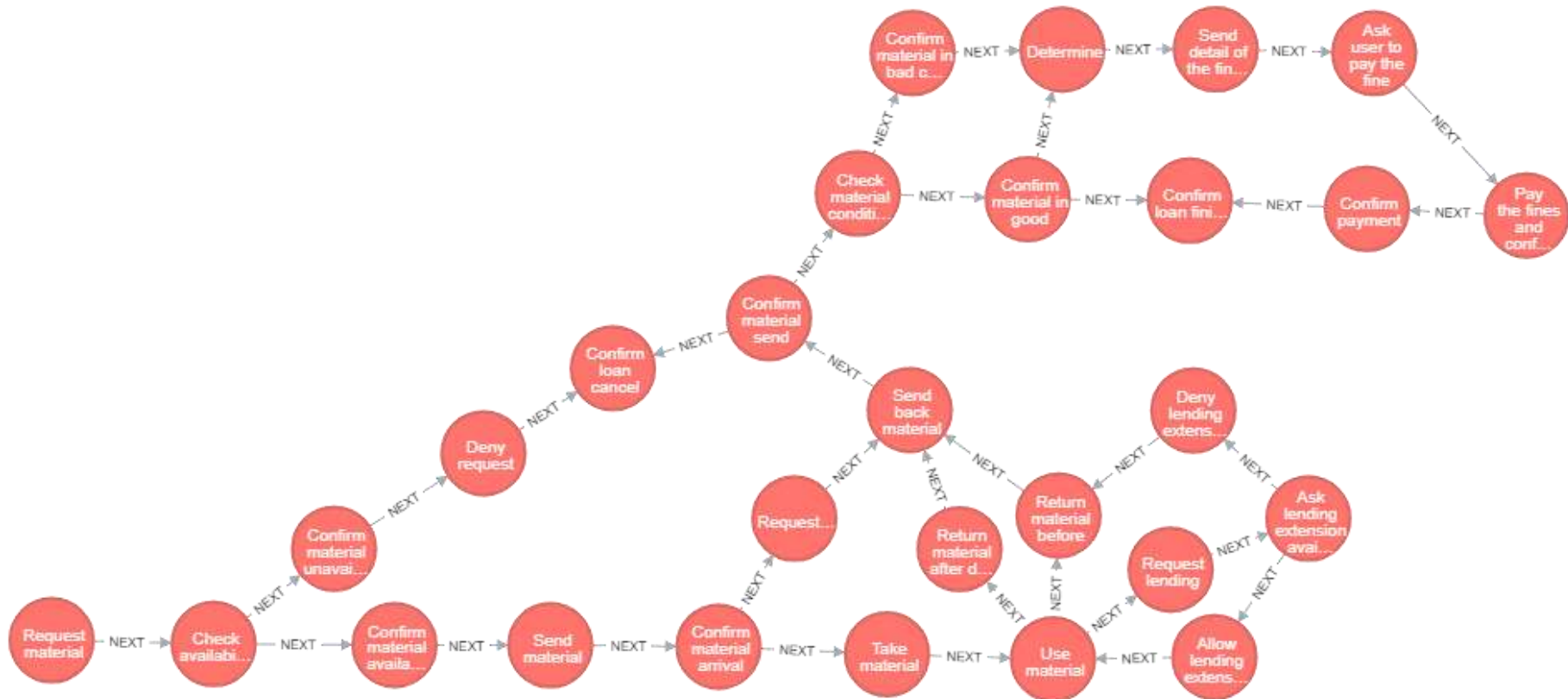
1. Menguji apakah proses bisnis yang terbentuk dari *event log* yang telah dibuat sudah menangani *non-free choice* dengan benar.
2. Menguji apakah proses bisnis yang terbentuk dari *event log* yang mengandung *invisible task* sudah menangani *non-free choice* dengan benar.
3. Menguji apakah proses bisnis yang terbentuk dari *event log* yang datanya telah disaring sudah menangani *non-free-choice* dengan benar.
4. Membandingkan metode manakah yang terbaik untuk memodelkan proses bisnis.

5.3 Hasil Implementasi Pembentukan Model Bisnis Proses menggunakan Metode Pertama

Pada bab ini akan dijelaskan hasil implemetasi dari pembentukan model proses bisnis. Proses model yang terbentuk terbagi menjadi 2 yaitu event log asli dan event log yang telah ditambahkan *invisible task*.

5.3.1 Hasil Implementasi Pembentukan Model Bisnis Proses dari Event Log

Awalnya event log dibuat dalam format *xlsx* yaitu format asli Microsoft Excel diubah menjadi format *csv*. *Event log* yang telah diubah menjadi format *csv* di load kedalam *neo4j*. Load data dilakukan dua kali dan masing-masing load data dimasukkan ke dalam label yang berbeda. Label yang terbentuk adalah label *Activity* dan label *CaseActivity*. Keseluruhan tahapan selanjutnya akan dilakukan menggunakan label *CaseActivity*. Kemudian hasil graf yang telah di load ditambahkan relasi *sequence* dalam kasus ini relasi *sequence* digambarkan oleh NEXT seperti pada Gambar 5.1.



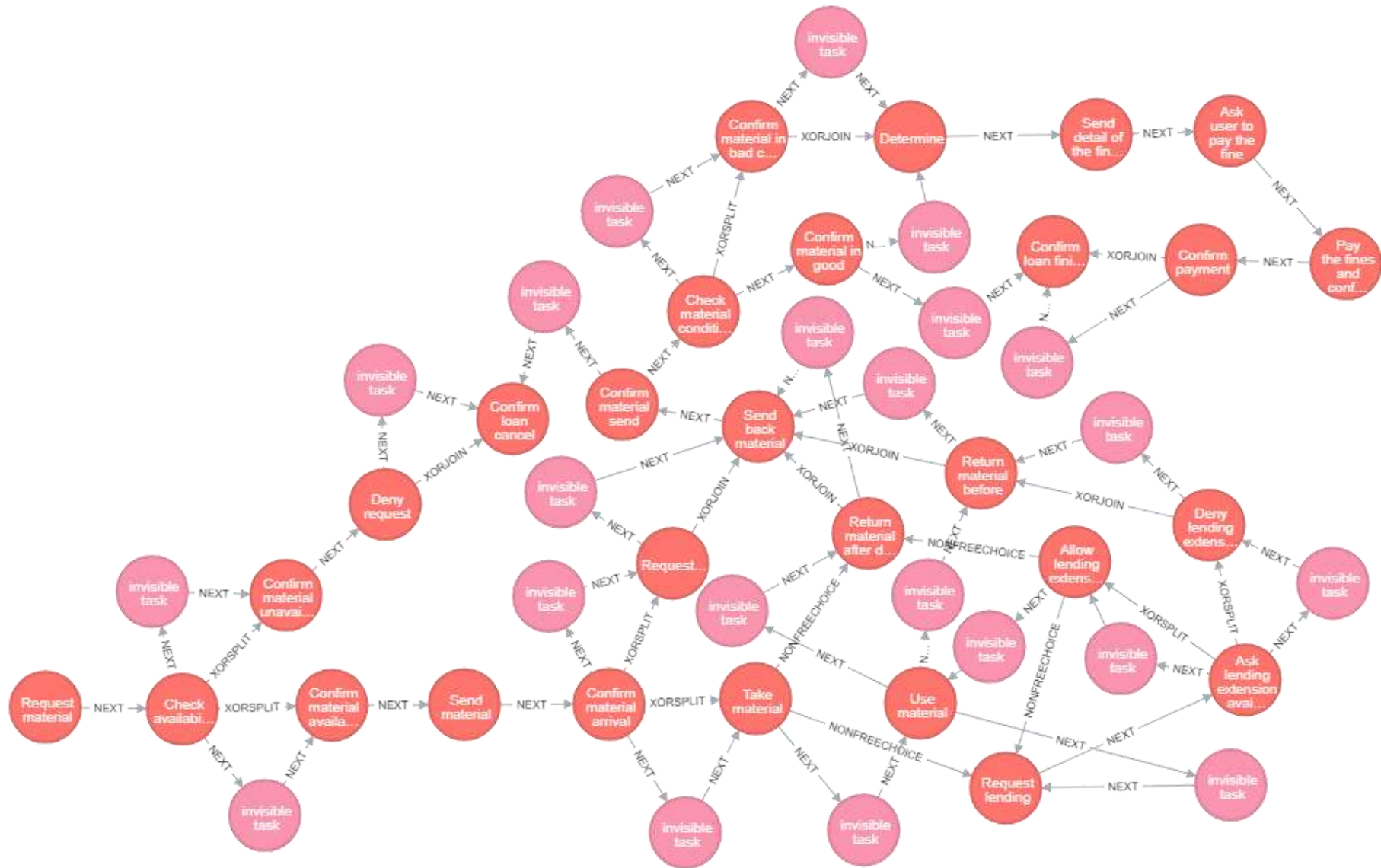
Gambar 5.1 Hasil Penemuan *Sequence Event Log* pada Graf Neo4J

[Halaman ini sengaja dikosongkan]

choice berupa relasi. Pada Gambar 5.4 ditemukan 4 relasi NONFREECHOICE pada graf yang telah terbentuk sebelumnya.



Gambar 5.3 Hasil Penemuan *Invisible Task* pada Graf Neo4j



Gambar 5.4 Hasil Penemuan *Non-Free-Choice* pada Graf Neo4j

[Halaman ini sengaja dikosongkan]

5.4 Hasil Implementasi Pembentukan Model Bisnis Proses Menggunakan Metode Kedua

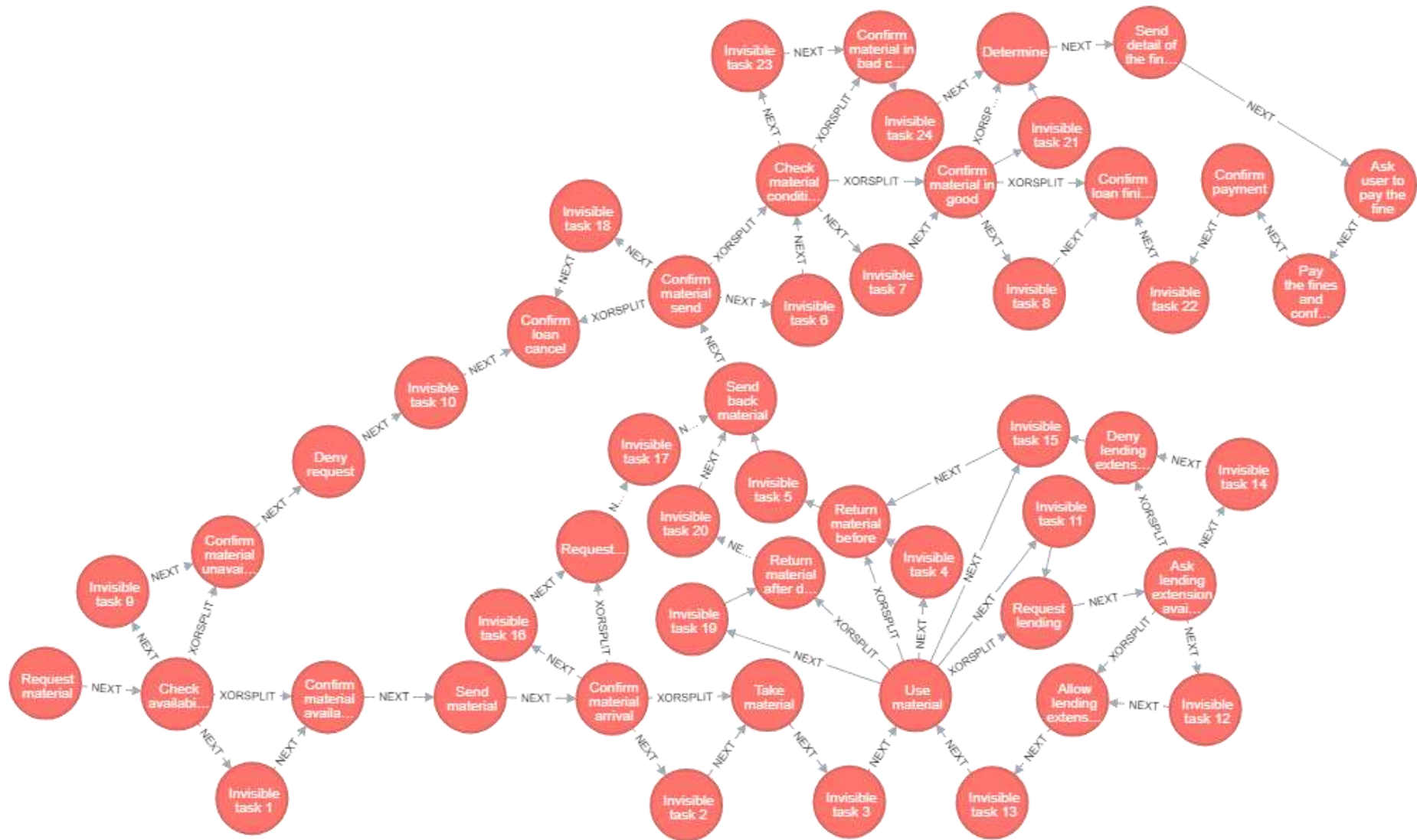
Tabel 5.1 Event Log Sebelum (Kiri) dan Setelah (Kanan) ditambahkan Invisible Task

CaseId	Activity	CaseId	Activity
PP1	Request material	PP1	Request material
PP1	Check availability	PP1	Check availability
PP1	Confirm material availability	PP1	Invisible task 1
PP1	Send material	PP1	Confirm material availability
PP1	Confirm material arrival	PP1	Send material
PP1	Take material	PP1	Confirm material arrival
PP1	Use material	PP1	Invisible task 2
PP1	Return material before due time	PP1	Take material
PP1	Send back material	PP1	Invisible task 3
PP1	Confirm material send back arrival	PP1	Use material
PP1	Check material condition	PP1	Invisible task 4
PP1	Confirm material in good condition	PP1	Return material before due time
PP1	Confirm loan finish	PP1	Invisible task 5
		PP1	Send back material
		PP1	Confirm material send back arrival
		PP1	Invisible task 6
		PP1	Check material condition
		PP1	Invisible task 7
		PP1	Confirm material in good condition
		PP1	Invisible task 8
		PP1	Confirm loan finish

Tabel 5.1 adalah contoh event log ILL yang telah ditambahkan *invisible task* secara eksplisit. Setelah semua *invisible task* telah ditambahkan ke dalam event log, langkah selanjutnya sama dengan implementasi sebelumnya, langkah-langkah yang dilakukan

adalah load atau import *event log* ke Neo4J, kemudian tambahkan penemuan *sequence* dan *control-flow pattern XOR* ke graf yang telah terbentuk. Graf yang telah terbentuk dapat dilihat pada Gambar 5.5.

Setelah melakukan penemuan *sequence* dan XOR, maka lakukan load *event log* sebagai *sequence* untuk melakukan penemuan *non-free-choice* pada *event log*. Kemudian lakukan penemuan *non-free-choice* dengan membandingkan model proses bisnis *sequence* dengan model proses bisnis biasa. Tetapi tidak ditemukan relasi *non-free-choice* pada metode ini.



Gambar 5.5 Hasil Pemodelan Proses Bisnis dengan *Event Log* yang ditambahkan *Invisible Task* pada Graf Neo4J

[Halaman ini sengaja dikosongkan]

5.5 Hasil Implementasi Pembentukan Model Bisnis Proses Menggunakan Metode Ketiga

Gambar 7.3 pada halaman lampiran merupakan hasil dari *load* data dan penemuan *sequence* tanpa penggabungan *node*. Berdasarkan metode ini terbentuk 1268 node dan 1381 relasi *sequence* dari 70 case. Gambar 5.6 merupakan model proses bisnis yang sama dengan Gambar 7.3 tetapi dengan node dibatasi hanya menggunakan dua *case* yaitu *caseId* PP1 dan PP2.



Gambar 5.6 Hasil Pemodelan Proses Bisnis Tanpa Penggabungan *Node* dengan *CaseId* PP1 dan PP2

5.5.1 Hasil Implementasi *Trace Clustering*, Penyaringan Data Anomali, dan Penggabungan *Node*

Frequency	Flow type	CaseId
12	[Request material, "Check availability", "Confirm material availability", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock", "Check material in stock", "Confirm material in stock", "Check material in stock", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock"]	[PP1, PP2] PP1, PP2, PP1 PP1, PP1, PP1 PP1, PP1, PP1 PP1
8	[Request material, "Check availability", "Confirm material availability", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock", "Check material in stock", "Confirm material in stock", "Check material in stock", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock"]	[PP1, PP2] PP1, PP1, PP1 PP1, PP1, PP1 PP1, PP1, PP1 PP1
8	[Request material, "Check availability", "Confirm material availability", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock", "Check material in stock", "Confirm material in stock", "Check material in stock", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock"]	[PP1, PP2] PP1, PP1, PP1 PP1, PP1, PP1 PP1, PP1, PP1 PP1
8	[Request material, "Check availability", "Confirm material availability", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock", "Check material in stock", "Confirm material in stock", "Check material in stock", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock"]	[PP1, PP2] PP1, PP1, PP1 PP1, PP1, PP1 PP1, PP1, PP1 PP1
8	[Request material, "Check availability", "Confirm material availability", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock", "Check material in stock", "Confirm material in stock", "Check material in stock", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock"]	[PP1, PP2] PP1, PP1, PP1 PP1, PP1, PP1 PP1, PP1, PP1 PP1
8	[Request material, "Check availability", "Confirm material availability", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock", "Check material in stock", "Confirm material in stock", "Check material in stock", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock"]	[PP1, PP2] PP1, PP1, PP1 PP1, PP1, PP1 PP1, PP1, PP1 PP1
8	[Request material, "Check availability", "Confirm material availability", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock", "Check material in stock", "Confirm material in stock", "Check material in stock", "Send material", "Confirm material arrival", "Take material", "Use material", "Return material to stock"]	[PP1, PP2] PP1, PP1, PP1 PP1, PP1, PP1 PP1, PP1, PP1 PP1

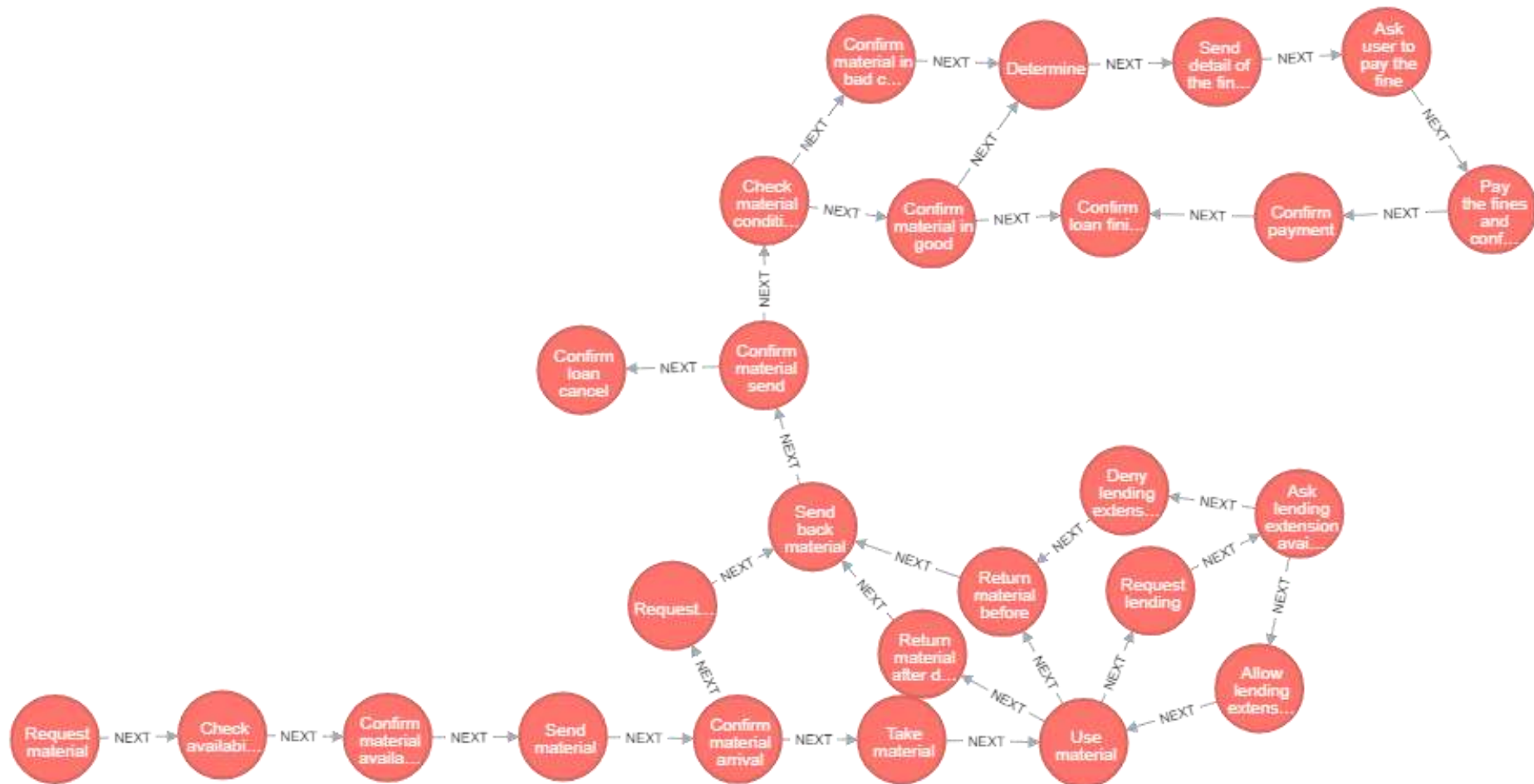
Gambar 5.7 Hasil *Trace Clustering* Menggunakan Neo4J

Setelah load dan penemuan *sequence* tanpa penggabungan *node*, langkah selanjutnya adalah menampilkan daftar *case* dan *trace* apa saja yang ada di *event log*. Gambar 5.7 merupakan tampilan *trace clustering* di Neo4J dan Tabel 7.1 pada halaman lampiran merupakan hasil *export trace clustering* ke Microsoft Excel. Pada langkah ini ditemukan bahwa CaseId PP2, PP21, dan PP53 memiliki jumlah penggunaan atau pengulangan paling sedikit. Sehingga diputuskan bahwa 3 CaseId tersebut yang akan dihapus nodenya dari graf yang telah terbentuk.

Setelah melakukan perhitungan panjang *trace* seperti pada Gambar 5.8 maka data yang perlu dihapus adalah data yang memiliki *trace* sebanyak 5, karena CaseId PP2, PP21, dan PP53 masing-masing memiliki aktivitas sebanyak 6. Data hasil penghitungan panjang *Trace* secara keluruhan dapat dilihat Tabel 7.3 pada halaman lampiran. Setelah menghapus data langkah selanjutnya adalah menghapus relasi NEXT atau melakukan penggabungan data. Kedua langkah ini dapat dilakukan secara tidak urut. Kemudian langkah selanjutnya adalah melakukan penemuan *sequence* normal.

CaseId	Case	Activities	Activities_Count
5	[PP17, PP52, PP51, PP12, PP53]	[Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Make material, Use material, Request loading container, Not loading container availability, Make loading container, Use material, Return material before due time, Send back material, Confirm material and back arrival, Check material arrival, Confirm material is back condition, Transceive the amount of flow to be paid, Send detail of the flow, Not use to pay the flow, Pay the flow and confirm payment, Confirm payment, Confirm date limit]	22
6	[PP18, PP19, PP54, PP11, PP55]	[Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Make material, Use material, Request loading container, Not loading container availability, Make loading container, Use material, Return material after due time, Send back material, Confirm material and back arrival, Check material arrival, Confirm material is good condition, Transceive the amount of flow to be paid, Send detail of the flow, Not use to pay the flow, Pay the flow and confirm payment, Confirm payment, Confirm date limit]	22
8	[PP56, PP11, PP59, PP11, PP55]	[Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Make material, Use material, Request loading container, Not loading container availability, Make loading container, Use material, Return material before due time, Send back material, Confirm material and back arrival, Check material arrival, Confirm material is good condition, Confirm date limit, Confirm date limit]	21
8	[PP54, PP12, PP17, PP11, PP56, PP11]	[Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Make material, Use material, Request loading container, Not loading container availability, Make loading container, Return material before due time, Send back material, Confirm material and back arrival, Check material arrival, Confirm material is good condition, Transceive the amount of flow to be paid, Send detail of the flow, Not use to pay the flow, Pay the flow and confirm payment, Confirm payment, Confirm date limit]	21
8	[PP56, PP17, PP12, PP55]	[Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Make material, Use material, Request loading container, Not loading container availability, Make loading container, Use material, Return material before due time, Send back material, Confirm material and back arrival, Check material arrival, Confirm material is good condition, Transceive the amount of flow to be paid, Send detail of the flow, Not use to pay the flow, Pay the flow and confirm payment, Confirm payment, Confirm date limit]	20

Gambar 5.8 Hasil Penghitungan Panjang Trace dalam Setiap Case Menggunakan Neo4J



Gambar 5.9 Hasil Penyaringan Data dan Penemuan *Sequence Event Log* pada Graf Neo4J

[Halaman ini sengaja dikosongkan]

Dapat dilihat pada Gambar 5.9 tidak terdapat aktivitas yang hanya ada pada caseId PP2, PP21, dan PP53. Aktivitas yang telah dihapus adalah Confirm material unavailable dan Deny request. Kemudian seperti pada graf 1, langkah selanjutnya adalah penemuan *control-flow pattern XOR* pada Gambar 5.10.



Gambar 5.10 Hasil Penemuan Control-Flow Pattern pada Graf Neo4J

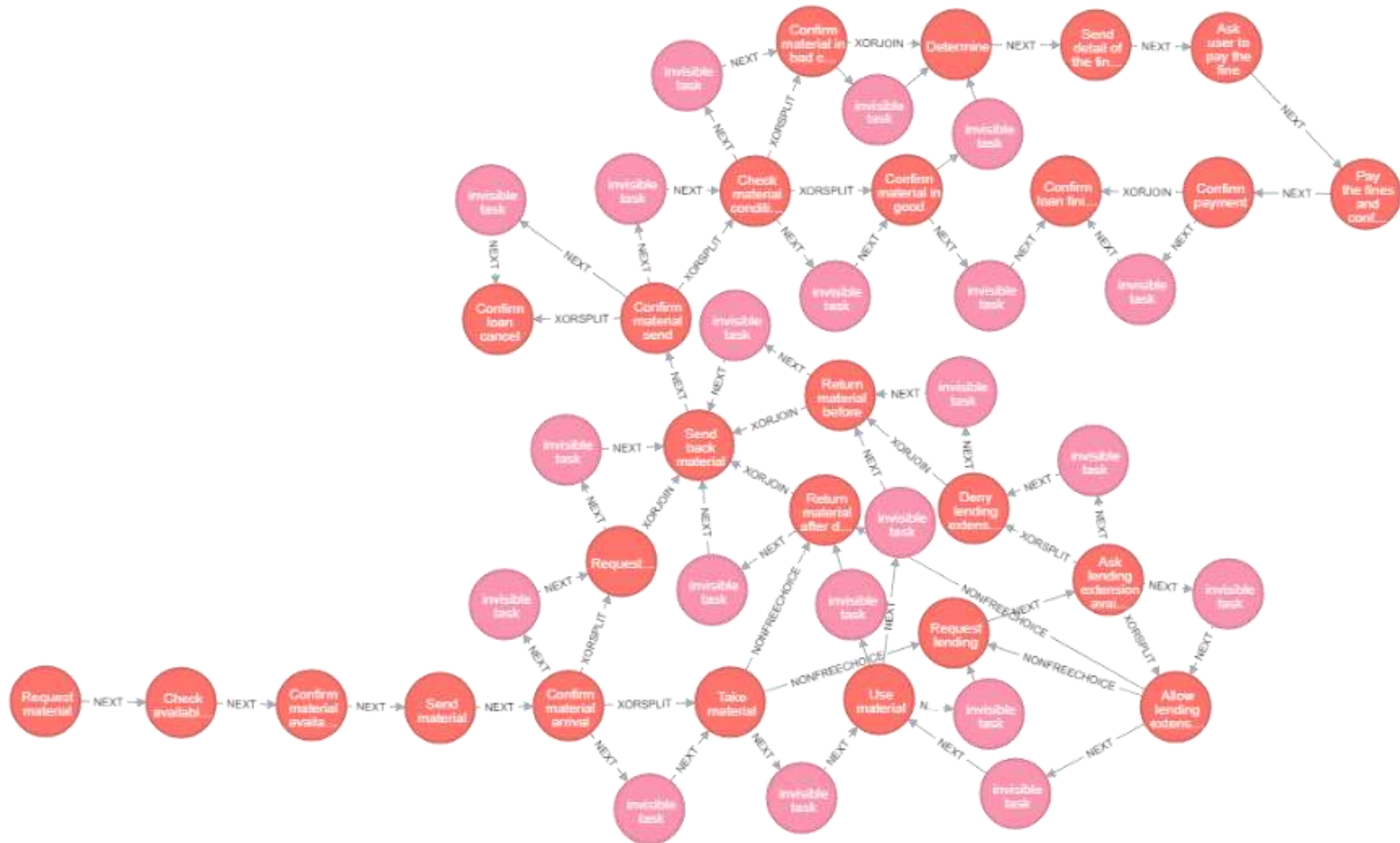
5.5.2 Hasil Penemuan Invisible Task dan Non-Free-Choice

Gambar 5.12 adalah hasil dari penemuan *invisible task* dan *non-free-choice*. *Invisible task* di representasikan dengan node berwarna merah muda, sedangkan *non-free-choice* direpresentasikan dengan garis relasi bernama NONFREECHOICE. Pada metode ketiga, jumlah relasi NONFREECHOICE yang ditemukan sebanyak 4.

Gambar 5.11 adalah hasil pengelompokan relasi *sequence* antar aktivitas menggunakan Neo4J, dengan menggunakan pengelompokan relasi, pengguna dapat melihat jumlah relasi antar node, hal ini tidak dapat dilakukan menggunakan metode pertama. Daftar lengkap relasi antar node atau aktivitas dapat dilihat pada Tabel 7.2 di halaman lampiran.

Frequency	a.Name	Relation	c.Name
15	"Allow lending extension"	NEXT	"Use machine"
48	"Ask lending extension availability"	NEXT	"Allow lending extension"
22	"Ask lending extension availability"	NEXT	"Deny lending extension"
12	"Ask material availability"	NEXT	"Check availability"
32	"Ask user to pay the fine"	NEXT	"Pay fine fees and confirm payment"
67	"Check availability"	NEXT	"Confirm material availability"
3	"Check availability"	NEXT	"Confirm material unavailable"
64	"Check material condition"	NEXT	"Confirm material is good condition"
18	"Check material condition"	NEXT	"Confirm material is bad condition"
6	"Confirm material arrival"	NEXT	"Request material pickup around flow line"
68	"Confirm material arrival"	NEXT	"Take material"
67	"Confirm material availability"	NEXT	"Send material"
18	"Confirm material is bad condition"	NEXT	"Determine the amount of fines to be paid"
31	"Confirm material is good condition"	NEXT	"Confirm loan held"
12	"Confirm material is good condition"	NEXT	"Determine the amount of fines to be paid"
6	"Confirm material sent back arrival"	NEXT	"Confirm loan cancel"
68	"Confirm material sent back arrival"	NEXT	"Check material condition"
3	"Confirm material unavailable"	NEXT	"Deny request"
36	"Confirm payment"	NEXT	"Confirm loan held"
16	"Deny lending extension"	NEXT	"Return material before due time"
3	"Deny request"	NEXT	"Confirm loan cancel"

Gambar 5.11 Hasil Pengelompokan Relasi *Sequence* Tiap Aktivitas Menggunakan Neo4J



Gambar 5.12 Hasil Penemuan *Non-Free-Choice* pada Graf Neo4J

[Halaman ini sengaja dikosongkan]

5.6 Perbandingan Hasil Pemodelan Proses Bisnis

Berdasarkan ketiga graf yang telah terbentuk pada Gambar 5.4, Gambar 5.5, dan Gambar 5.12, beberapa perbedaan secara umum dapat dilihat dari *node* dan relasi ketiga graf. Graf pertama dan ketiga mengandung *non-free-choice*, sedangkan *non-free-choice* tidak ditemukan pada graf kedua. Graf ketiga memiliki dua node lebih sedikit dibandingkan graf pertama dan kedua, graf ketiga memiliki 27 node aktivitas, sedangkan graf pertama dan kedua memiliki 29 node aktivitas.

Gambar 5.13 adalah hasil pemodelan proses bisnis menggunakan ProM dengan *event log* Interlibrary Loan, Gambar 5.13 merepresentasikan graf pada metode pertama. Gambar 5.14 adalah hasil pemodelan proses bisnis menggunakan ProM dengan *event log interlibrary loan* yang telah ditambahkan *invisible task*, Gambar 5.14 merepresentasikan graf pada metode kedua. Gambar 5.15 adalah hasil pemodelan proses bisnis menggunakan ProM dengan *event log* Interlibrary Loan yang telah disaring sebelumnya sesuai dengan hasil dari penyaringan data di Neo4J, Gambar 5.15 merepresentasikan graf pada metode ketiga.

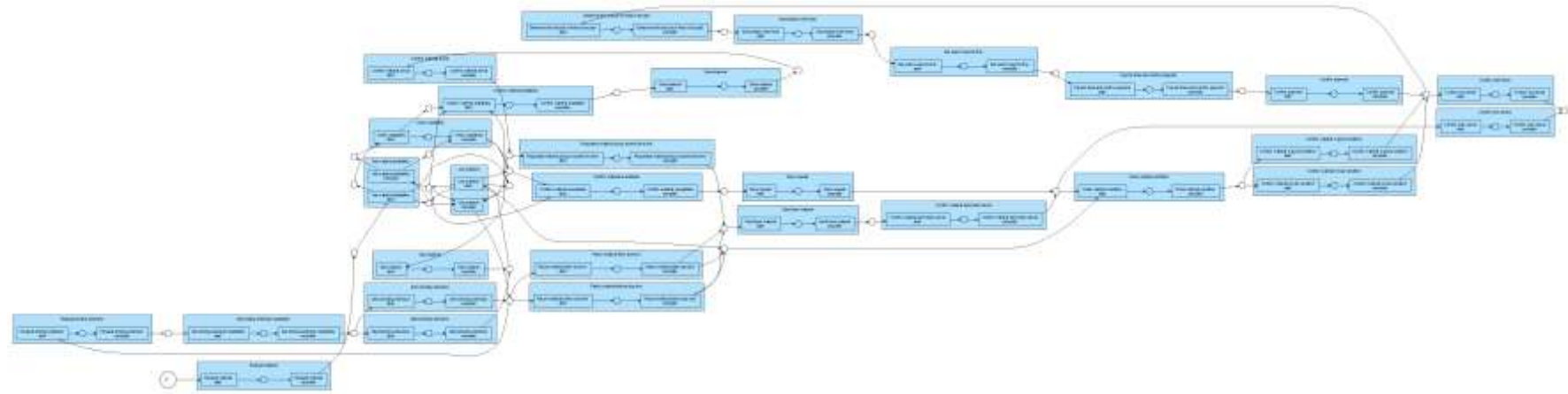
Dapat dilihat pada Gambar 5.13, Gambar 5.14, dan Gambar 5.15 tidak terdapat *non-free-choice*, sehingga dapat disimpulkan bahwa graf kedua lebih benar dibandingkan dengan graf pertama dan ketiga. Kesalahan dapat disebabkan oleh algoritma penemuan *non-free-choice*, sehingga perlu dilakukan perbaikan untuk pengimplementasian selanjutnya.

Pemodelan menggunakan metode pertama lebih mudah dipelajari, karena hanya membutuhkan 5 langkah dalam prosesnya. Tetapi metode pertama memiliki beberapa kekurangan, yaitu tidak bisa melakukan trace clustering, melihat jumlah relasi antar node, dan tidak bisa menghapus atau menyaring data anomali.

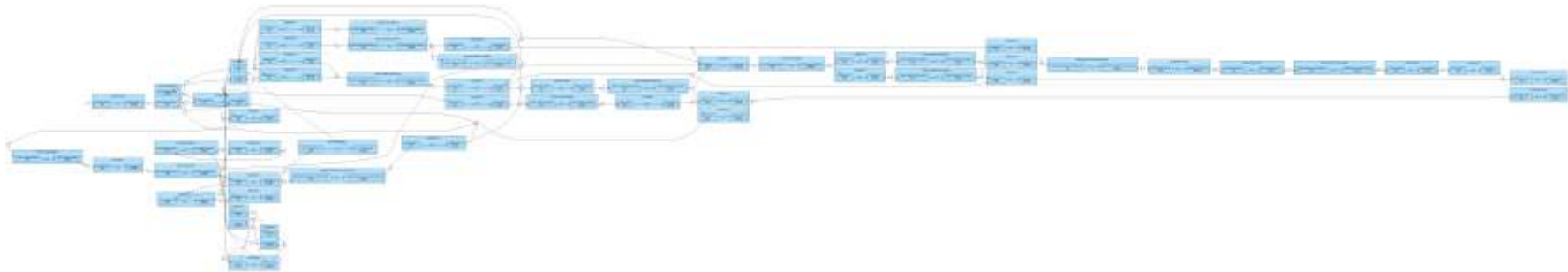
Pemodelan menggunakan metode kedua merupakan satu-satunya metode yang dapat menggambarkan *invisible task* dan *non-free-choice* menggunakan ProM. Penemuan *invisible task* tidak bisa dilakukan di ProM, karena tidak ada algoritma $\alpha\#$ pada program ProM. Kekurangan dari metode ini adalah pengguna harus

menambahkan sendiri *invisible task* pada *event log*, sehingga memakan waktu yang banyak apabila tidak menggunakan program bantuan pada saat *preprocessing event log*.

Pemodelan memiliki langkah pengerjaan lebih banyak dari kedua metode sebelumnya. Kelebihan dari metode ini yaitu pengguna dapat memodelkan proses bisnis lebih sesuai dengan kebutuhan pengguna. Metode ketiga ini juga bisa digunakan untuk melihat jumlah relasi antar node pada proses bisnis, sehingga memungkinkan penerapan atau implementasi algoritma heuristic menggunakan metode ini.

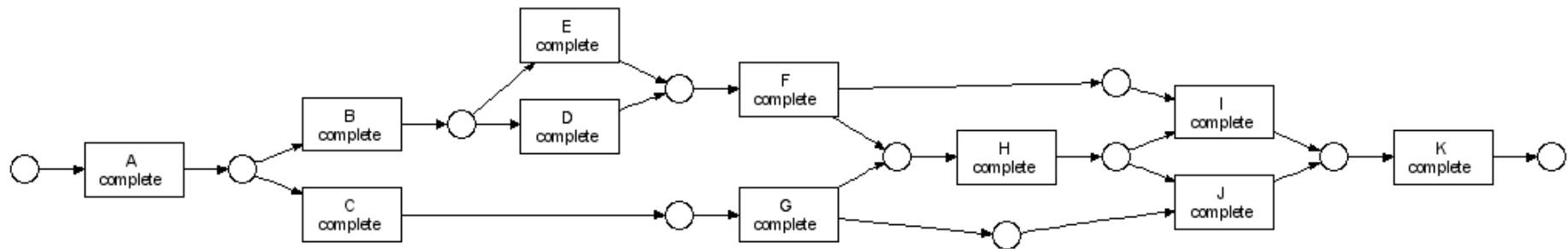


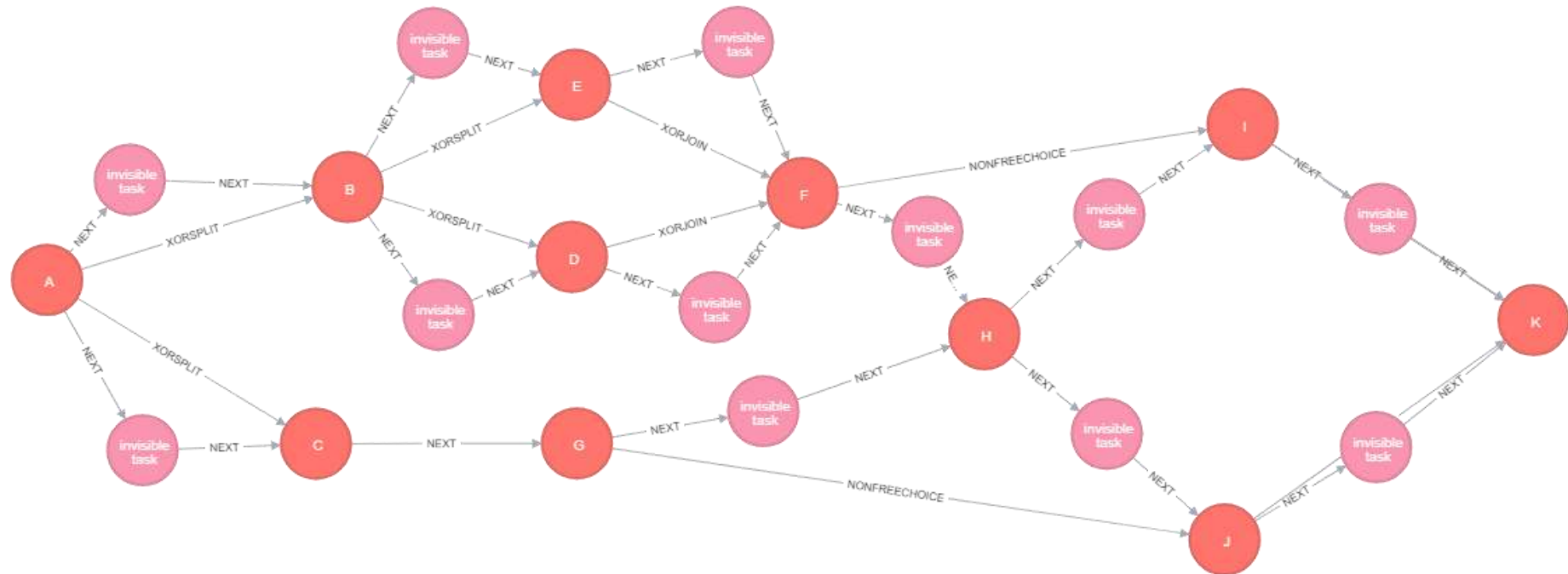
Gambar 5.13 Hasil Pemodelan Proses Bisnis *Interlibrary Loan* Menggunakan ProM



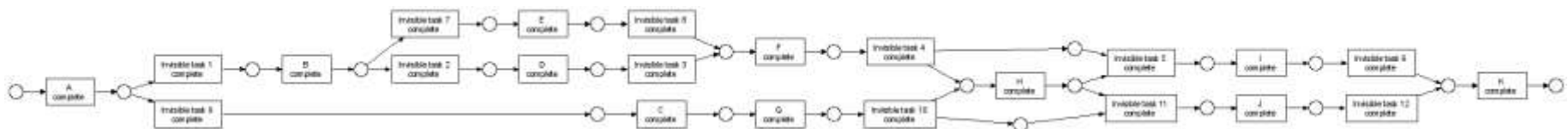
Gambar 5.14 Hasil Pemodelan Proses Bisnis *Interlibrary Loan* dengan *Event Log* yang Mengandung *Invisible Task* Menggunakan ProM

Gambar 5.16 Hasil Pemodelan Proses Bisnis yang Mengandung *Non-Free-Choice* Menggunakan ProM

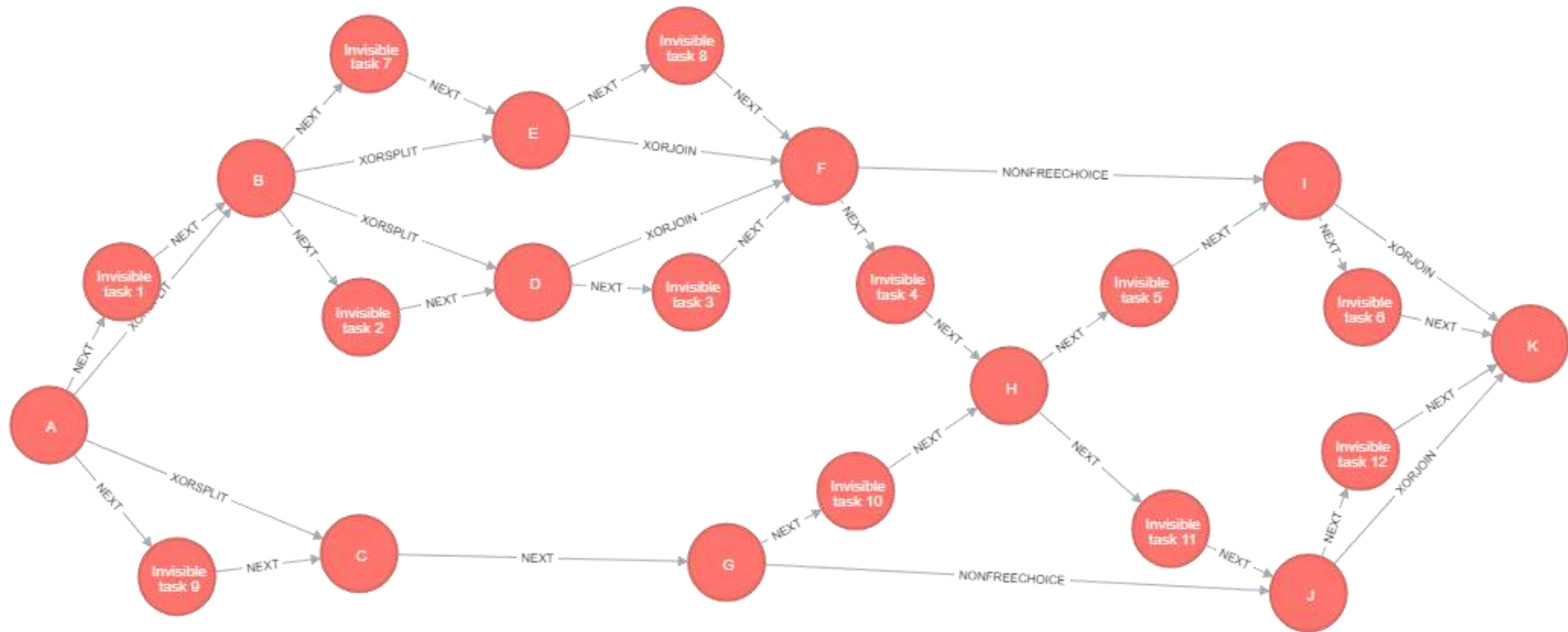




Gambar 5.17 Hasil Pemodelan Proses Bisnis yang Mengandung *Non-Free-Choice* Menggunakan Neo4J



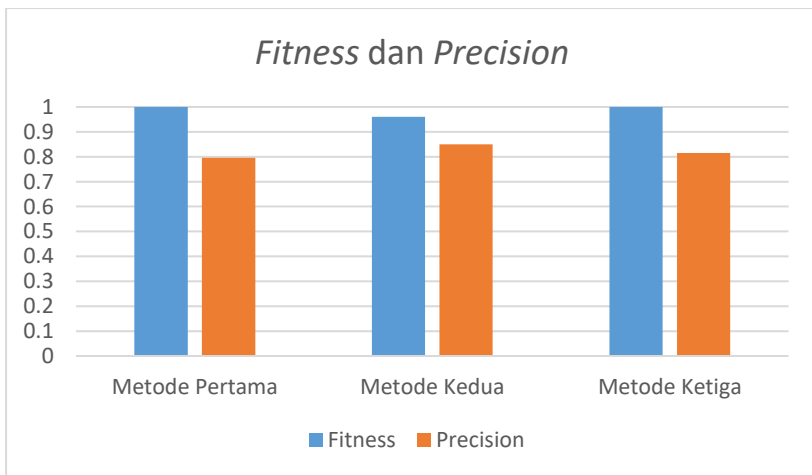
Gambar 5.18 Hasil Pemodelan Proses Bisnis dengan *Event Log* yang Mengandung *Invisible Task* dan *Non-Free-Choice* Menggunakan ProM



Gambar 5.19 Hasil Pemodelan Proses Bisnis dengan *Event Log* yang Mengandung *Invisible Task* dan *Non-Free-Choice* Menggunakan Neo4J

Gambar 5.16 merupakan hasil penemuan *event log* yang mengandung *non-free-choice* menggunakan ProM. Terlihat bahwa relasi *non-free-choice* terbentuk pada aktivitas F dengan I dan aktivitas G dengan J. Gambar 5.17 merupakan proses penemuan *event log* menggunakan Neo4J. *Event log* yang digunakan pada Gambar 5.17 sama dengan *event log* yang digunakan pada Gambar 5.16. Hasil pemodelan proses bisnis menggunakan Neo4J juga telah berhasil menemukan *non-free-choice* sama dengan hasil pada penemuan di ProM. Perbedaan diantara kedua hasil penemuan diatas adalah Neo4J berhasil menemukan *invisible task*, sedangkan *invisible task* tidak dapat ditemukan menggunakan ProM.

Gambar 5.18 merupakan hasil penemuan *event log* yang telah ditambahkan *invisible task* dan mengandung *non-free-choice* menggunakan ProM. Gambar 5.19 merupakan hasil penemuan *event log* yang telah ditambahkan *invisible task* dan mengandung *non-free-choice* menggunakan Neo4J. Dapat dilihat pada Gambar 5.19 dan Gambar 5.18 kedua pemodelan tersebut berhasil menemukan *invisible task* dan *non-free-choice* pada *event log* menggunakan ProM dan Neo4J. Hasil *non-free-choice* pada Gambar 5.18 dan 5.25 sama dengan hasil *non-free-choice* pada Gambar 5.17 dan Gambar 5.17 yaitu antara aktivitas F dengan I dan aktivitas G dengan J.



Gambar 5.20 Hasil *Fitness* dan *Precision* untuk Setiap Metode

Gambar 5.20 adalah hasil fitness dan precision dari ketiga metode yang digunakan pada Tugas Akhir ini. Metode pertama dan metode memiliki fitness yang sama, yaitu 1. Metode kedua memiliki fitness yang lebih rendah dengan fitness sebesar 0.959974. Metode kedua memiliki precision yang paling tinggi sebesar 0.850122. Metode ketiga memiliki precision yang lebih tinggi dari metode pertama dengan besar 0.81481093, sedangkan metode pertama memiliki precision sebesar 0.7959951.

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan Tugas Akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Model proses bisnis yang benar dapat dibuat dengan cara melakukan penemuan *invisible task* dan *non-free-choice* pada satu model proses bisnis. Langkah-langkah penemuan model proses bisnis pada Neo4J adalah import data *event log* ke dalam Neo4J, kemudian implementasikan algoritma sesuai pada Tabel 3.3 untuk menemukan relasi *sequence*, relasi *control-flow pattern*, node-node *invisible task*, dan relasi *non-free-choice*.
2. Model proses bisnis yang benar dan dapat digunakan pada Neo4J maupun ProM dapat dibuat dengan cara melakukan pemodelan proses bisnis dengan menggunakan *event log* yang telah ditambahkan *invisible task*. *Invisible task* ditambahkan ke dalam *event log* sesuai dengan peraturan pada subbab 3.5.
3. Penyaringan data anomali dapat dilakukan dengan *trace clustering* dan penghapusan data yang berfrekuensi paling rendah, dalam kasus interlibrary loan data yang berfrekuensi paling rendah adalah 3. Model proses bisnis yang datanya telah disaring memiliki hasil *precision* yang lebih tinggi yaitu sebesar 0.81481093 jika dibandingkan dengan *precision* model proses bisnis yang tidak di saring datanya yaitu sebesar 0.7959951.

6.2 Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Perbaiki algoritma penemuan *non-free-choice*, agar bisa mendeteksi *non-free-choice* apabila banyak cabang pada model proses bisnis.
2. Pembuatan perangkat lunak untuk memudahkan penambahan *invisible task* pada *event log*, sehingga pengguna tidak perlu menambahkan *invisible task* secara manual.
3. SOP yang dianjurkan adalah hasil graf model pada Gambar 5.1 karena sudah mencakup semua *case request* atau permintaan peminjaman material *Interlibrary Loan*.

DAFTAR PUSTAKA

- [1] *Pennsylvania Interlibrary Loan Procedures Manual*, Pennsylvania: Harrisburg: Institute of Museum and Library Services, 2011.
- [2] I. Robinson, J. Webber and E. Eifrem, *Graph Databases: New Opportunities for Connected Data*, O'Reilly Media, Inc., 2015.
- [3] W. M. P. van der Aalst, *Process Mining - Discovery, Conformance and Enhancement of Business Processes*, Schleiden: Springer, 2010.
- [4] A. A. K. Medeiros, B. F. van Dongen, W. M. P. van der Aalst and A. J. M. M. Weijters, "Process mining : extending the alpha-algorithm to mine short loops," Technische Universiteit Eindhoven, Eindhoven, 2004.
- [5] L. Wen, W. van der Aalst and J. e. a. Wang, "Mining Process Models with Non-Free-Choice Constructs," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 145-180, 2007.
- [6] L. Wen, J. Wang, W. van der Aalst, B. Huang and J. Sun, "Mining process models with prime invisible tasks," *Data & Knowledge Engineering* 69, pp. 999-1021, 2010.
- [7] Q. Guo, L. Wen, J. Wang, Z. Yan and P. S. Yu, "Mining Invisible Tasks in Non-free-choice Constructs," in *Business Process Management. BPM 2016.*, Rio de Janeiro, 2015.
- [8] U. o. C. S. B. Library, "Interlibrary Loan (ILL) | UCSB Library," [Online]. Available: <https://www.library.ucsb.edu/interlibrary-loan>. [Accessed 4 January 2018].
- [9] R. Sarno, Y. Effendi and F. Haryadita, "Modified Time-Based Heuristics Miner," *IRECOS*, vol. 14, no. 1, 2016.

- [10] A. Weijters, W. van der Aalst and A. de Medeiros, "Process mining with the Heuristics Miner-algorithm," Eindhoven University of Technology, Netherlands, 2006.
- [11] A. Y. Hadiwijaya, R. Sarno and D. Sunaryono, Business Process Intelligence berdasarkan Graph Model (Studi Kasus : Enterprise Resource Planning Retail), Surabaya, 2018.
- [12] R. Johanes, R. Sarno and D. Sunaryono, Memperbaiki Business Process Mengandung Invisible Task Menggunakan Graph Model, Surabaya, 2018.
- [13] K. R. Sungkono and R. Sarno, "Constructing Control-Flow Patterns Containing Invisible Task and Non-Free Choice," *International Journal of Innovative Computing, Information and Control (IJICIC)*, vol. 14, no. 4, p. 1285–1299, August 2018.
- [14] M. Song, C. W. Gunther and W. M. P. van der Aalst, "Trace Clustering in Process Mining," in *Business Process Management Workshops. BPM 2008*, Milan, 2008.

LAMPIRAN

Tabel 7.1 Hasil Trace Clustering Menggunakan Neo4J

Freq.	Trace Type	CaseId
12	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Confirm loan finish	PP34, PP16, PP1, PP48, PP15, PP18, PP68, PP50, PP47, PP20, PP67, PP52
8	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Return material after due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Determine the amounts of fines to be paid, Send detail of the fines, Ask user to pay the fine, Pay the fines and confirm payment, Confirm payment, Confirm loan finish	PP49, PP58, PP7, PP19, PP39, PP51, PP26, PP17
6	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Allow lending extension, Use material, Request lending extension, Ask lending extension availability, Deny lending extension, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Confirm loan finish	PP25, PP57, PP69, PP6, PP38, PP70
6	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Deny lending extension, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in bad condition, Determine the amounts of fines to be paid, Send detail of the fines, Ask user to pay the fine, Pay the fines and confirm payment, Confirm payment, Confirm loan finish	PP31, PP13, PP45, PP12, PP44, PP32
5	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Allow lending extension, Use material, Return material after due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Determine the amounts of fines to be paid, Send detail of	PP10, PP63, PP66, PP42, PP29

Freq.	Trace Type	CaseId
	the fines, Ask user to pay the fine, Pay the fines and confirm payment, Confirm payment, Confirm loan finish	
5	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Allow lending extension, Use material, Request lending extension, Ask lending extension availability, Allow lending extension, Use material, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Confirm loan finish	PP46, PP33, PP62, PP65, PP14
5	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Allow lending extension, Use material, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in bad condition, Determine the amounts of fines to be paid, Send detail of the fines, Ask user to pay the fine, Pay the fines and confirm payment, Confirm payment, Confirm loan finish	PP61, PP43, PP64, PP30, PP11
4	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Requested material pickup exceed time limit, Send back material, Confirm material send back arrival, Confirm loan cancel	PP37, PP24, PP56, PP5
4	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Deny lending extension, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Confirm loan finish	PP55, PP4, PP36, PP23
4	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Allow lending extension, Use material, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Confirm loan finish	PP22, PP3, PP54, PP35
4	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Return material after due time, Send back material, Confirm material send back arrival,	PP28, PP9, PP60, PP41

Freq.	Trace Type	CaseId
	Check material condition, Confirm material in bad condition, Determine the amounts of fines to be paid, Send detail of the fines, Ask user to pay the fine, Pay the fines and confirm payment, Confirm payment, Confirm loan finish	
4	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in bad condition, Determine the amounts of fines to be paid, Send detail of the fines, Ask user to pay the fine, Pay the fines and confirm payment, Confirm payment, Confirm loan finish	PP40, PP27, PP59, PP8
3	Request material, Check availability, Confirm material unavailable, Deny request, Confirm loan cancel	PP21, PP53, PP2

Tabel 7.2 Hasil Pengelompokan Relasi *Sequence* Tiap Aktivitas

Frek.	Aktivitas Asal	Relasi	Aktivitas Tujuan
70	Allow lending extension	NEXT	Use material
46	Ask lending extension availability	NEXT	Allow lending extension
22	Ask lending extension availability	NEXT	Deny lending extension
32	Ask user to pay the fine	NEXT	Pay the fines and confirm payment
67	Check availability	NEXT	Confirm material availability
3	Check availability	NEXT	Confirm material unavailable
44	Check material condition	NEXT	Confirm material in good condition
19	Check material condition	NEXT	Confirm material in bad condition
4	Confirm material arrival	NEXT	Requested material pickup exceed time limit
63	Confirm material arrival	NEXT	Take material
67	Confirm material availability	NEXT	Send material
19	Confirm material in bad condition	NEXT	Determine the amounts of fines to be paid
31	Confirm material in good condition	NEXT	Confirm loan finish
13	Confirm material in good condition	NEXT	Determine the amounts of fines to be paid
4	Confirm material send back arrival	NEXT	Confirm loan cancel

Frek.	Aktivitas Asal	Relasi	Aktivitas Tujuan
63	Confirm material send back arrival	NEXT	Check material condition
3	Confirm material unavailable	NEXT	Deny request
32	Confirm payment	NEXT	Confirm loan finish
16	Deny lending extension	NEXT	Return material before due time
3	Deny request	NEXT	Confirm loan cancel
32	Determine the amounts of fines to be paid	NEXT	Send detail of the fines
32	Pay the fines and confirm payment	NEXT	Confirm payment
68	Request lending extension	NEXT	Ask lending extension availability
70	Request material	NEXT	Check availability
4	Requested material pickup exceed time limit	NEXT	Send back material
17	Return material after due time	NEXT	Send back material
46	Return material before due time	NEXT	Send back material
67	Send back material	NEXT	Confirm material send back arrival
32	Send detail of the fines	NEXT	Ask user to pay the fine
67	Send material	NEXT	Confirm material arrival
93	Take material	NEXT	Use material
92	Use material	NEXT	Request lending extension
22	Use material	NEXT	Return material after due time
49	Use material	NEXT	Return material before due time

Tabel 7.3 Hasil Penghitungan Panjang Trace dalam Setiap Case

Freq.	CaseId	Trace Type	Trace Length
5	PP11, PP64, PP61, PP43, PP30	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Allow lending extension, Use material, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in bad condition, Determine the amounts of fines to be paid, Send detail of the fines, Ask user to pay the fine, Pay the fines and confirm payment, Confirm payment, Confirm loan finish	22
5	PP29, PP10, PP63,	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending	22

Freq.	Caseld	Trace Type	Trace Length
	PP42, PP66	extension, Ask lending extension availability, Allow lending extension, Use material, Return material after due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Determine the amounts of fines to be paid, Send detail of the fines, Ask user to pay the fine, Pay the fines and confirm payment, Confirm payment, Confirm loan finish	
5	PP65, PP14, PP46, PP33, PP62	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Allow lending extension, Use material, Request lending extension, Ask lending extension availability, Allow lending extension, Use material, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Confirm loan finish	21
6	PP44, PP32, PP31, PP13, PP45, PP12	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Deny lending extension, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in bad condition, Determine the amounts of fines to be paid, Send detail of the fines, Ask user to pay the fine, Pay the fines and confirm payment, Confirm payment, Confirm loan finish	21
6	PP38, PP70, PP25, PP69, PP57, PP6	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Allow lending extension, Use material, Request lending extension, Ask lending extension availability, Deny lending extension, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Confirm loan finish	20
4	PP59, PP8, PP40, PP27	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in bad condition, Determine the amounts of fines to be paid, Send detail of the fines, Ask user to	18

Freq.	Caseld	Trace Type	Trace Length
		pay the fine, Pay the fines and confirm payment, Confirm payment, Confirm loan finish	
4	PP28, PP60, PP9, PP41	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Return material after due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in bad condition, Determine the amounts of fines to be paid, Send detail of the fines, Ask user to pay the fine, Pay the fines and confirm payment, Confirm payment, Confirm loan finish	18
8	PP17, PP26, PP58, PP49, PP19, PP7, PP51, PP39	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Return material after due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Determine the amounts of fines to be paid, Send detail of the fines, Ask user to pay the fine, Pay the fines and confirm payment, Confirm payment, Confirm loan finish	18
4	PP22, PP54, PP35, PP3	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Allow lending extension, Use material, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Confirm loan finish	17
4	PP23, PP55, PP4, PP36	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Request lending extension, Ask lending extension availability, Deny lending extension, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Confirm loan finish	16
12	PP50, PP47, PP20, PP52, PP67, PP34, PP16, PP48, PP15, PP18,	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Take material, Use material, Return material before due time, Send back material, Confirm material send back arrival, Check material condition, Confirm material in good condition, Confirm loan finish	13

Freq.	CaseId	Trace Type	Trace Length
	PP68, PP1		
4	PP5, PP37, PP24, PP56	Request material, Check availability, Confirm material availability, Send material, Confirm material arrival, Requested material pickup exceed time limit, Send back material, Confirm material send back arrival, Confirm loan cancel	9
3	PPP2, PP21, PP53	Request material, Check availability, Confirm material unavailable, Deny request, Confirm loan cancel	5

Tabel 7.4 Event Log Interlibrary Loan

CaseId	Activity	Start Timestamp	End Timestamp
PP1	Request material	02-05-17 07:05	02-05-17 07:10
PP1	Check availability	02-05-17 07:10	02-05-17 10:15
PP1	Confirm material availability	02-05-17 10:15	02-05-17 10:20
PP1	Send material	02-05-17 10:20	02-05-17 14:15
PP1	Confirm material arrival	02-05-17 14:15	02-05-17 14:20
PP1	Take material	02-05-17 14:20	03-05-17 14:35
PP1	Use material	03-05-17 14:35	15-05-17 07:15
PP1	Return material before due time	15-05-17 07:15	15-05-17 07:30
PP1	Send back material	15-05-17 07:30	15-05-17 10:07
PP1	Confirm material send back arrival	15-05-17 10:07	15-05-17 10:12
PP1	Check material condition	15-05-17 10:12	15-05-17 13:12
PP1	Confirm material in good condition	15-05-17 13:12	15-05-17 13:17
PP1	Confirm loan finish	15-05-17 13:17	15-05-17 13:25
PP2	Request material	16-05-17 07:05	16-05-17 07:10
PP2	Check availability	16-05-17 07:10	18-05-17 07:15
PP2	Confirm material unavailable	18-05-17 07:15	18-05-17 07:20
PP2	Deny request	18-05-17 07:20	18-05-17 07:25
PP2	Confirm loan cancel	18-05-17 07:25	18-05-17 07:30
PP3	Request material	19-05-17 07:05	19-05-17 07:10
PP3	Check availability	19-05-17 07:10	19-05-17 10:15
PP3	Confirm material availability	19-05-17 10:15	19-05-17 10:20
PP3	Send material	19-05-17 10:20	19-05-17 13:20
PP3	Confirm material arrival	19-05-17 13:20	19-05-17 13:25
PP3	Take material	19-05-17 13:25	19-05-17 13:35
PP3	Use material	19-05-17 13:35	31-05-17 09:05
PP3	Request lending extension	31-05-17 09:05	31-05-17 09:15
PP3	Ask lending extension availability	31-05-17 09:15	31-05-17 12:15
PP3	Allow lending extension	31-05-17 12:15	31-05-17 12:20
PP3	Use material	31-05-17 12:20	05-06-17 08:20
PP3	Return material before due time	05-06-17 08:20	05-06-17 08:35
PP3	Send back material	05-06-17 08:35	05-06-17 13:35

CaseId	Activity	Start Timestamp	End Timestamp
PP3	Confirm material send back arrival	05-06-17 13:35	05-06-17 13:40
PP3	Check material condition	05-06-17 13:40	05-06-17 15:00
PP3	Confirm material in good condition	05-06-17 15:00	05-06-17 15:05
PP3	Confirm loan finish	05-06-17 15:05	05-06-17 15:10
PP4	Request material	06-06-17 07:05	06-06-17 07:10
PP4	Check availability	06-06-17 07:10	06-06-17 10:15
PP4	Confirm material availability	06-06-17 10:15	06-06-17 10:20
PP4	Send material	06-06-17 10:20	06-06-17 14:20
PP4	Confirm material arrival	06-06-17 14:20	06-06-17 14:25
PP4	Take material	06-06-17 14:25	06-06-17 14:35
PP4	Use material	06-06-17 14:35	19-06-17 14:35
PP4	Request lending extension	19-06-17 14:35	19-06-17 14:50
PP4	Ask lending extension availability	19-06-17 14:50	19-06-17 14:55
PP4	Deny lending extension	19-06-17 14:55	19-06-17 15:10
PP4	Return material before due time	19-06-17 15:10	19-06-17 15:20
PP4	Send back material	20-06-17 07:20	20-06-17 13:20
PP4	Confirm material send back arrival	20-06-17 13:20	20-06-17 13:25
PP4	Check material condition	20-06-17 13:25	20-06-17 15:25
PP4	Confirm material in good condition	20-06-17 15:25	20-06-17 15:30
PP4	Confirm loan finish	20-06-17 15:30	20-06-17 15:35
PP5	Request material	21-06-17 07:05	21-06-17 07:10
PP5	Check availability	21-06-17 07:10	21-06-17 10:15
PP5	Confirm material availability	21-06-17 10:15	21-06-17 10:20
PP5	Send material	21-06-17 10:20	21-06-17 15:25
PP5	Confirm material arrival	21-06-17 15:25	21-06-17 15:30
PP5	Requested material pickup exceed time limit	21-06-17 15:30	28-06-17 15:30
PP5	Send back material	29-06-17 07:05	29-06-17 11:05
PP5	Confirm material send back arrival	29-06-17 11:05	29-06-17 11:10
PP5	Confirm loan cancel	29-06-17 11:10	29-06-17 11:15
PP6	Request material	30-06-17 07:05	30-06-17 07:10
PP6	Check availability	30-06-17 07:10	30-06-17 10:15
PP6	Confirm material availability	30-06-17 10:15	30-06-17 10:20
PP6	Send material	30-06-17 10:20	30-06-17 13:20
PP6	Confirm material arrival	30-06-17 13:20	30-06-17 13:25
PP6	Take material	30-06-17 13:25	30-06-17 13:35
PP6	Use material	30-06-17 13:35	11-07-17 13:35
PP6	Request lending extension	11-07-17 13:35	11-07-17 13:45
PP6	Ask lending extension availability	11-07-17 13:45	11-07-17 13:50
PP6	Allow lending extension	11-07-17 13:50	11-07-17 14:20
PP6	Use material	11-07-17 14:20	23-07-17 14:20
PP6	Request lending extension	23-07-17 14:20	23-07-17 14:30
PP6	Ask lending extension availability	23-07-17 14:30	23-07-17 14:35
PP6	Deny lending extension	23-07-17 14:35	23-07-17 15:00
PP6	Return material before due time	23-07-17 15:00	23-07-17 15:10

CaseId	Activity	Start Timestamp	End Timestamp
PP6	Send back material	24-07-17 07:10	24-07-17 11:10
PP6	Confirm material send back arrival	24-07-17 11:10	24-07-17 11:15
PP6	Check material condition	24-07-17 11:15	24-07-17 14:15
PP6	Confirm material in good condition	24-07-17 14:15	24-07-17 14:20
PP6	Confirm loan finish	24-07-17 14:20	24-07-17 14:25
PP7	Request material	25-07-17 07:05	25-07-17 07:10
PP7	Check availability	25-07-17 07:10	25-07-17 10:15
PP7	Confirm material availability	25-07-17 10:15	25-07-17 10:20
PP7	Send material	25-07-17 10:20	25-07-17 15:20
PP7	Confirm material arrival	25-07-17 15:20	25-07-17 15:25
PP7	Take material	26-07-17 07:20	26-07-17 07:30
PP7	Use material	26-07-17 07:30	11-08-17 07:30
PP7	Return material after due time	11-08-17 07:30	11-08-17 07:40
PP7	Send back material	11-08-17 07:40	11-08-17 11:40
PP7	Confirm material send back arrival	11-08-17 11:40	11-08-17 11:45
PP7	Check material condition	11-08-17 11:45	11-08-17 14:00
PP7	Confirm material in good condition	11-08-17 14:00	11-08-17 14:05
PP7	Determine the amounts of fines to be paid	11-08-17 14:05	11-08-17 14:15
PP7	Send detail of the fines	11-08-17 14:15	11-08-17 14:20
PP7	Ask user to pay the fine	11-08-17 14:20	11-08-17 14:25
PP7	Pay the fines and confirm payment	11-08-17 14:25	11-08-17 16:25
PP7	Confirm payment	11-08-17 16:25	14-08-17 07:10
PP7	Confirm loan finish	14-08-17 07:10	14-08-17 07:15
PP8	Request material	15-08-17 07:05	15-08-17 07:10
PP8	Check availability	15-08-17 07:10	15-08-17 10:15
PP8	Confirm material availability	15-08-17 10:15	15-08-17 10:20
PP8	Send material	15-08-17 10:20	15-08-17 15:20
PP8	Confirm material arrival	15-08-17 15:20	15-08-17 15:25
PP8	Take material	16-08-17 07:20	16-08-17 07:30
PP8	Use material	16-08-17 07:30	23-08-17 07:30
PP8	Return material before due time	23-08-17 07:30	23-08-17 07:40
PP8	Send back material	23-08-17 07:40	23-08-17 12:40
PP8	Confirm material send back arrival	23-08-17 12:40	23-08-17 12:45
PP8	Check material condition	23-08-17 12:45	23-08-17 15:00
PP8	Confirm material in bad condition	23-08-17 15:00	23-08-17 15:05
PP8	Determine the amounts of fines to be paid	23-08-17 15:05	23-08-17 15:15
PP8	Send detail of the fines	23-08-17 15:15	23-08-17 15:25
PP8	Ask user to pay the fine	23-08-17 15:25	23-08-17 15:30
PP8	Pay the fines and confirm payment	23-08-17 15:30	24-08-17 07:30
PP8	Confirm payment	24-08-17 07:30	24-08-17 07:40
PP8	Confirm loan finish	24-08-17 07:40	24-08-17 07:45
PP9	Request material	25-08-17 07:05	25-08-17 07:10
PP9	Check availability	25-08-17 07:10	25-08-17 10:15

CaseId	Activity	Start Timestamp	End Timestamp
PP9	Confirm material availability	25-08-17 10:15	25-08-17 10:20
PP9	Send material	25-08-17 10:20	25-08-17 13:20
PP9	Confirm material arrival	25-08-17 13:20	25-08-17 13:25
PP9	Take material	25-08-17 13:25	25-08-17 13:40
PP9	Use material	25-08-17 13:40	11-09-17 07:40
PP9	Return material after due time	11-09-17 07:40	11-09-17 07:50
PP9	Send back material	11-09-17 07:50	11-09-17 11:00
PP9	Confirm material send back arrival	11-09-17 11:00	11-09-17 11:05
PP9	Check material condition	11-09-17 11:05	11-09-17 14:05
PP9	Confirm material in bad condition	11-09-17 14:05	11-09-17 14:10
PP9	Determine the amounts of fines to be paid	11-09-17 14:10	11-09-17 14:20
PP9	Send detail of the fines	11-09-17 14:20	11-09-17 14:30
PP9	Ask user to pay the fine	11-09-17 14:30	11-09-17 14:35
PP9	Pay the fines and confirm payment	11-09-17 14:35	11-09-17 17:40
PP9	Confirm payment	11-09-17 17:40	12-09-17 07:30
PP9	Confirm loan finish	12-09-17 07:30	12-09-17 07:35
PP10	Request material	13-09-17 07:05	13-09-17 07:10
PP10	Check availability	13-09-17 07:10	13-09-17 10:15
PP10	Confirm material availability	13-09-17 10:15	13-09-17 10:20
PP10	Send material	13-09-17 10:20	13-09-17 15:20
PP10	Confirm material arrival	13-09-17 15:20	13-09-17 15:25
PP10	Take material	14-09-17 07:10	14-09-17 07:20
PP10	Use material	14-09-17 07:20	25-09-17 07:20
PP10	Request lending extension	25-09-17 07:20	25-09-17 07:30
PP10	Ask lending extension availability	25-09-17 07:30	25-09-17 07:40
PP10	Allow lending extension	25-09-17 07:40	25-09-17 08:20
PP10	Use material	25-09-17 08:20	10-10-17 07:20
PP10	Return material after due time	10-10-17 07:20	10-10-17 07:30
PP10	Send back material	10-10-17 07:30	10-10-17 12:30
PP10	Confirm material send back arrival	10-10-17 12:30	10-10-17 12:35
PP10	Check material condition	10-10-17 12:35	10-10-17 14:50
PP10	Confirm material in good condition	10-10-17 14:50	10-10-17 14:55
PP10	Determine the amounts of fines to be paid	10-10-17 14:55	10-10-17 15:10
PP10	Send detail of the fines	10-10-17 15:10	10-10-17 15:20
PP10	Ask user to pay the fine	10-10-17 15:20	10-10-17 15:25
PP10	Pay the fines and confirm payment	10-10-17 15:25	10-10-17 20:25
PP10	Confirm payment	10-10-17 20:25	11-10-17 07:35
PP10	Confirm loan finish	11-10-17 07:35	11-10-17 07:40
PP11	Request material	12-10-17 07:05	12-10-17 07:10
PP11	Check availability	12-10-17 07:10	12-10-17 10:15
PP11	Confirm material availability	12-10-17 10:15	12-10-17 10:20
PP11	Send material	12-10-17 10:20	12-10-17 14:20
PP11	Confirm material arrival	12-10-17 14:20	12-10-17 14:25

CaseId	Activity	Start Timestamp	End Timestamp
PP11	Take material	12-10-17 14:25	12-10-17 14:40
PP11	Use material	12-10-17 14:40	23-10-17 07:50
PP11	Request lending extension	23-10-17 07:50	23-10-17 08:10
PP11	Ask lending extension availability	23-10-17 08:10	23-10-17 08:20
PP11	Allow lending extension	23-10-17 08:20	23-10-17 08:30
PP11	Use material	23-10-17 08:30	30-10-17 08:30
PP11	Return material before due time	30-10-17 08:30	30-10-17 08:40
PP11	Send back material	30-10-17 08:40	30-10-17 12:40
PP11	Confirm material send back arrival	30-10-17 12:40	30-10-17 12:45
PP11	Check material condition	30-10-17 12:45	30-10-17 14:45
PP11	Confirm material in bad condition	30-10-17 14:45	30-10-17 14:50
PP11	Determine the amounts of fines to be paid	30-10-17 14:50	30-10-17 15:05
PP11	Send detail of the fines	30-10-17 15:05	30-10-17 15:15
PP11	Ask user to pay the fine	30-10-17 15:15	30-10-17 15:20
PP11	Pay the fines and confirm payment	30-10-17 15:20	30-10-17 19:20
PP11	Confirm payment	30-10-17 19:20	31-10-17 07:35
PP11	Confirm loan finish	31-10-17 07:35	31-10-17 07:40
PP12	Request material	01-11-17 07:05	01-11-17 07:10
PP12	Check availability	01-11-17 07:10	01-11-17 10:15
PP12	Confirm material availability	01-11-17 10:15	01-11-17 10:20
PP12	Send material	01-11-17 10:20	01-11-17 13:50
PP12	Confirm material arrival	01-11-17 13:50	01-11-17 13:55
PP12	Take material	01-11-17 13:55	01-11-17 14:05
PP12	Use material	01-11-17 14:05	13-11-17 14:05
PP12	Request lending extension	13-11-17 14:05	13-11-17 14:15
PP12	Ask lending extension availability	13-11-17 14:15	13-11-17 14:25
PP12	Deny lending extension	13-11-17 14:25	13-11-17 14:50
PP12	Return material before due time	13-11-17 14:50	13-11-17 15:00
PP12	Send back material	14-11-17 07:00	14-11-17 10:00
PP12	Confirm material send back arrival	14-11-17 10:00	14-11-17 10:05
PP12	Check material condition	14-11-17 10:05	14-11-17 13:05
PP12	Confirm material in bad condition	14-11-17 13:05	14-11-17 13:10
PP12	Determine the amounts of fines to be paid	14-11-17 13:10	14-11-17 13:20
PP12	Send detail of the fines	14-11-17 13:20	14-11-17 13:25
PP12	Ask user to pay the fine	14-11-17 13:25	14-11-17 13:30
PP12	Pay the fines and confirm payment	14-11-17 13:30	14-11-17 15:30
PP12	Confirm payment	14-11-17 15:30	15-11-17 07:10
PP12	Confirm loan finish	15-11-17 07:10	15-11-17 07:15
PP13	Request material	29-11-17 07:15	29-11-17 07:20
PP13	Check availability	29-11-17 07:20	30-11-17 10:25
PP13	Confirm material availability	30-11-17 10:25	30-11-17 10:30
PP13	Send material	30-11-17 10:30	30-11-17 14:30
PP13	Confirm material arrival	30-11-17 14:30	30-11-17 14:35

CaseId	Activity	Start Timestamp	End Timestamp
PP13	Take material	30-11-17 14:35	30-11-17 14:35
PP13	Use material	30-11-17 14:35	10-12-17 07:20
PP13	Request lending extension	10-12-17 07:20	10-12-17 07:30
PP13	Ask lending extension availability	10-12-17 07:30	10-12-17 07:35
PP13	Allow lending extension	10-12-17 07:35	10-12-17 08:35
PP13	Use material	10-12-17 08:35	21-12-17 08:35
PP13	Request lending extension	21-12-17 08:35	21-12-17 08:45
PP13	Ask lending extension availability	21-12-17 08:45	21-12-17 08:50
PP13	Allow lending extension	21-12-17 08:50	21-12-17 09:50
PP13	Use material	21-12-17 09:50	03-01-18 07:50
PP13	Return material before due time	03-01-18 07:50	03-01-18 08:00
PP13	Send back material	03-01-18 08:00	03-01-18 11:30
PP13	Confirm material send back arrival	03-01-18 11:30	03-01-18 11:35
PP13	Check material condition	03-01-18 11:35	03-01-18 14:35
PP13	Confirm material in good condition	03-01-18 14:35	03-01-18 14:40
PP13	Confirm loan finish	03-01-18 14:40	03-01-18 14:45

Tabel 7.5 Event Log Interlibrary Loan yang Mengandung Invisible Task

CaseId	Activity	Start Timestamp	End Timestamp
PP1	Request material	02-05-17 07:05	02-05-17 07:10
PP1	Check availability	02-05-17 07:10	02-05-17 10:14
PP1	Invisible task 1	02-05-17 10:14	02-05-17 10:15
PP1	Confirm material availability	02-05-17 10:15	02-05-17 10:20
PP1	Send material	02-05-17 10:20	02-05-17 14:15
PP1	Confirm material arrival	02-05-17 14:15	02-05-17 14:19
PP1	Invisible task 2	02-05-17 14:19	02-05-17 14:20
PP1	Take material	02-05-17 14:20	03-05-17 14:34
PP1	Invisible task 3	03-05-17 14:34	03-05-17 14:35
PP1	Use material	03-05-17 14:35	15-05-17 07:14
PP1	Invisible task 4	15-05-17 07:14	15-05-17 07:15
PP1	Return material before due time	15-05-17 07:15	15-05-17 07:29
PP1	Invisible task 5	15-05-17 07:29	15-05-17 07:30
PP1	Send back material	15-05-17 07:30	15-05-17 10:07
PP1	Confirm material send back arrival	15-05-17 10:07	15-05-17 10:11
PP1	Invisible task 6	15-05-17 10:11	15-05-17 10:12
PP1	Check material condition	15-05-17 10:12	15-05-17 13:11
PP1	Invisible task 7	15-05-17 13:11	15-05-17 13:12
PP1	Confirm material in good condition	15-05-17 13:12	15-05-17 13:16
PP1	Invisible task 8	15-05-17 13:16	15-05-17 13:17
PP1	Confirm loan finish	15-05-17 13:17	15-05-17 13:25
PP2	Request material	16-05-17 07:05	16-05-17 07:10
PP2	Check availability	16-05-17 07:10	18-05-17 07:14
PP2	Invisible task 9	18-05-17 07:14	18-05-17 07:15
PP2	Confirm material unavailable	18-05-17 07:15	18-05-17 07:20
PP2	Deny request	18-05-17 07:20	18-05-17 07:24

CaseId	Activity	Start_Timestamp	End_Timestamp
PP2	Invisible task 10	18-05-17 07:24	18-05-17 07:25
PP2	Confirm loan cancel	18-05-17 07:25	18-05-17 07:30
PP3	Request material	19-05-17 07:05	19-05-17 07:10
PP3	Check availability	19-05-17 07:10	19-05-17 10:14
PP3	Invisible task 1	19-05-17 10:14	19-05-17 10:15
PP3	Confirm material availability	19-05-17 10:15	19-05-17 10:20
PP3	Send material	19-05-17 10:20	19-05-17 13:20
PP3	Confirm material arrival	19-05-17 13:20	19-05-17 13:24
PP3	Invisible task 2	19-05-17 13:24	19-05-17 13:25
PP3	Take material	19-05-17 13:25	19-05-17 13:34
PP3	Invisible task 3	19-05-17 13:34	19-05-17 13:35
PP3	Use material	19-05-17 13:35	31-05-17 09:04
PP3	Invisible task 11	31-05-17 09:04	31-05-17 09:05
PP3	Request lending extension	31-05-17 09:05	31-05-17 09:15
PP3	Ask lending extension availability	31-05-17 09:15	31-05-17 12:14
PP3	Invisible task 12	31-05-17 12:14	31-05-17 12:15
PP3	Allow lending extension	31-05-17 12:15	31-05-17 12:19
PP3	Invisible task 13	31-05-17 12:19	31-05-17 12:20
PP3	Use material	31-05-17 12:20	05-06-17 08:19
PP3	Invisible task 4	05-06-17 08:19	05-06-17 08:20
PP3	Return material before due time	05-06-17 08:20	05-06-17 08:34
PP3	Invisible task 5	05-06-17 08:34	05-06-17 08:35
PP3	Send back material	05-06-17 08:35	05-06-17 13:35
PP3	Confirm material send back arrival	05-06-17 13:35	05-06-17 13:39
PP3	Invisible task 6	05-06-17 13:39	05-06-17 13:40
PP3	Check material condition	05-06-17 13:40	05-06-17 14:59
PP3	Invisible task 7	05-06-17 14:59	05-06-17 15:00
PP3	Confirm material in good condition	05-06-17 15:00	05-06-17 15:04
PP3	Invisible task 8	05-06-17 15:04	05-06-17 15:05
PP3	Confirm loan finish	05-06-17 15:05	05-06-17 15:10
PP4	Request material	06-06-17 07:05	06-06-17 07:10
PP4	Check availability	06-06-17 07:10	06-06-17 10:14
PP4	Invisible task 1	06-06-17 10:14	06-06-17 10:15
PP4	Confirm material availability	06-06-17 10:15	06-06-17 10:20
PP4	Send material	06-06-17 10:20	06-06-17 14:20
PP4	Confirm material arrival	06-06-17 14:20	06-06-17 14:24
PP4	Invisible task 2	06-06-17 14:24	06-06-17 14:25
PP4	Take material	06-06-17 14:25	06-06-17 14:34
PP4	Invisible task 3	06-06-17 14:34	06-06-17 14:35
PP4	Use material	06-06-17 14:35	19-06-17 14:34
PP4	Invisible task 11	19-06-17 14:34	19-06-17 14:35
PP4	Request lending extension	19-06-17 14:35	19-06-17 14:50
PP4	Ask lending extension availability	19-06-17 14:50	19-06-17 14:54
PP4	Invisible task 14	19-06-17 14:54	19-06-17 14:55
PP4	Deny lending extension	19-06-17 14:55	19-06-17 15:09

CaseId	Activity	Start_Timestamp	End_Timestamp
PP4	Invisible task 15	19-06-17 15:09	19-06-17 15:10
PP4	Return material before due time	19-06-17 15:10	20-06-17 07:19
PP4	Invisible task 5	20-06-17 07:19	20-06-17 07:20
PP4	Send back material	20-06-17 07:20	20-06-17 13:20
PP4	Confirm material send back arrival	20-06-17 13:20	20-06-17 13:24
PP4	Invisible task 6	20-06-17 13:24	20-06-17 13:25
PP4	Check material condition	20-06-17 13:25	20-06-17 15:24
PP4	Invisible task 7	20-06-17 15:24	20-06-17 15:25
PP4	Confirm material in good condition	20-06-17 15:25	20-06-17 15:29
PP4	Invisible task 8	20-06-17 15:29	20-06-17 15:30
PP4	Confirm loan finish	20-06-17 15:30	20-06-17 15:35
PP5	Request material	21-06-17 07:05	21-06-17 07:10
PP5	Check availability	21-06-17 07:10	21-06-17 10:14
PP5	Invisible task 1	21-06-17 10:14	21-06-17 10:15
PP5	Confirm material availability	21-06-17 10:15	21-06-17 10:20
PP5	Send material	21-06-17 10:20	21-06-17 15:25
PP5	Confirm material arrival	21-06-17 15:25	21-06-17 15:29
PP5	Invisible task 16	21-06-17 15:29	21-06-17 15:30
PP5	Requested material pickup exceed time limit	21-06-17 15:30	29-06-17 07:04
PP5	Invisible task 17	29-06-17 07:04	29-06-17 07:05
PP5	Send back material	29-06-17 07:05	29-06-17 11:05
PP5	Confirm material send back arrival	29-06-17 11:05	29-06-17 11:09
PP5	Invisible task 18	29-06-17 11:09	29-06-17 11:10
PP5	Confirm loan cancel	29-06-17 11:10	29-06-17 11:15
PP6	Request material	30-06-17 07:05	30-06-17 07:10
PP6	Check availability	30-06-17 07:10	30-06-17 10:14
PP6	Invisible task 1	30-06-17 10:14	30-06-17 10:15
PP6	Confirm material availability	30-06-17 10:15	30-06-17 10:20
PP6	Send material	30-06-17 10:20	30-06-17 13:20
PP6	Confirm material arrival	30-06-17 13:20	30-06-17 13:24
PP6	Invisible task 2	30-06-17 13:24	30-06-17 13:25
PP6	Take material	30-06-17 13:25	30-06-17 13:34
PP6	Invisible task 3	30-06-17 13:34	30-06-17 13:35
PP6	Use material	30-06-17 13:35	11-07-17 13:34
PP6	Invisible task 11	11-07-17 13:34	11-07-17 13:35
PP6	Request lending extension	11-07-17 13:35	11-07-17 13:45
PP6	Ask lending extension availability	11-07-17 13:45	11-07-17 13:49
PP6	Invisible task 12	11-07-17 13:49	11-07-17 13:50
PP6	Allow lending extension	11-07-17 13:50	11-07-17 14:19
PP6	Invisible task 13	11-07-17 14:19	11-07-17 14:20
PP6	Use material	11-07-17 14:20	23-07-17 14:19
PP6	Invisible task 11	23-07-17 14:19	23-07-17 14:20
PP6	Request lending extension	23-07-17 14:20	23-07-17 14:30
PP6	Ask lending extension availability	23-07-17 14:30	23-07-17 14:34

CaseId	Activity	Start_Timestamp	End_Timestamp
PP6	Invisible task 14	23-07-17 14:34	23-07-17 14:35
PP6	Deny lending extension	23-07-17 14:35	23-07-17 14:59
PP6	Invisible task 15	23-07-17 14:59	23-07-17 15:00
PP6	Return material before due time	23-07-17 15:00	24-07-17 07:09
PP6	Invisible task 5	24-07-17 07:09	24-07-17 07:10
PP6	Send back material	24-07-17 07:10	24-07-17 11:10
PP6	Confirm material send back arrival	24-07-17 11:10	24-07-17 11:14
PP6	Invisible task 6	24-07-17 11:14	24-07-17 11:15
PP6	Check material condition	24-07-17 11:15	24-07-17 14:14
PP6	Invisible task 7	24-07-17 14:14	24-07-17 14:15
PP6	Confirm material in good condition	24-07-17 14:15	24-07-17 14:19
PP6	Invisible task 8	24-07-17 14:19	24-07-17 14:20
PP6	Confirm loan finish	24-07-17 14:20	24-07-17 14:25
PP7	Request material	25-07-17 07:05	25-07-17 07:10
PP7	Check availability	25-07-17 07:10	25-07-17 10:14
PP7	Invisible task 1	25-07-17 10:14	25-07-17 10:15
PP7	Confirm material availability	25-07-17 10:15	25-07-17 10:20
PP7	Send material	25-07-17 10:20	25-07-17 15:20
PP7	Confirm material arrival	25-07-17 15:20	26-07-17 07:19
PP7	Invisible task 2	26-07-17 07:19	26-07-17 07:20
PP7	Take material	26-07-17 07:20	26-07-17 07:29
PP7	Invisible task 3	26-07-17 07:29	26-07-17 07:30
PP7	Use material	26-07-17 07:30	11-08-17 07:29
PP7	Invisible task 19	11-08-17 07:29	11-08-17 07:30
PP7	Return material after due time	11-08-17 07:30	11-08-17 07:39
PP7	Invisible task 20	11-08-17 07:39	11-08-17 07:40
PP7	Send back material	11-08-17 07:40	11-08-17 11:40
PP7	Confirm material send back arrival	11-08-17 11:40	11-08-17 11:44
PP7	Invisible task 6	11-08-17 11:44	11-08-17 11:45
PP7	Check material condition	11-08-17 11:45	11-08-17 13:59
PP7	Invisible task 7	11-08-17 13:59	11-08-17 14:00
PP7	Confirm material in good condition	11-08-17 14:00	11-08-17 14:04
PP7	Invisible task 21	11-08-17 14:04	11-08-17 14:05
PP7	Determine the amounts of fines to be paid	11-08-17 14:05	11-08-17 14:15
PP7	Send detail of the fines	11-08-17 14:15	11-08-17 14:20
PP7	Ask user to pay the fine	11-08-17 14:20	11-08-17 14:25
PP7	Pay the fines and confirm payment	11-08-17 14:25	11-08-17 16:25
PP7	Confirm payment	11-08-17 16:25	14-08-17 07:09
PP7	Invisible task 22	14-08-17 07:09	14-08-17 07:10
PP7	Confirm loan finish	14-08-17 07:10	14-08-17 07:15
PP8	Request material	15-08-17 07:05	15-08-17 07:10
PP8	Check availability	15-08-17 07:10	15-08-17 10:14
PP8	Invisible task 1	15-08-17 10:14	15-08-17 10:15
PP8	Confirm material availability	15-08-17 10:15	15-08-17 10:20

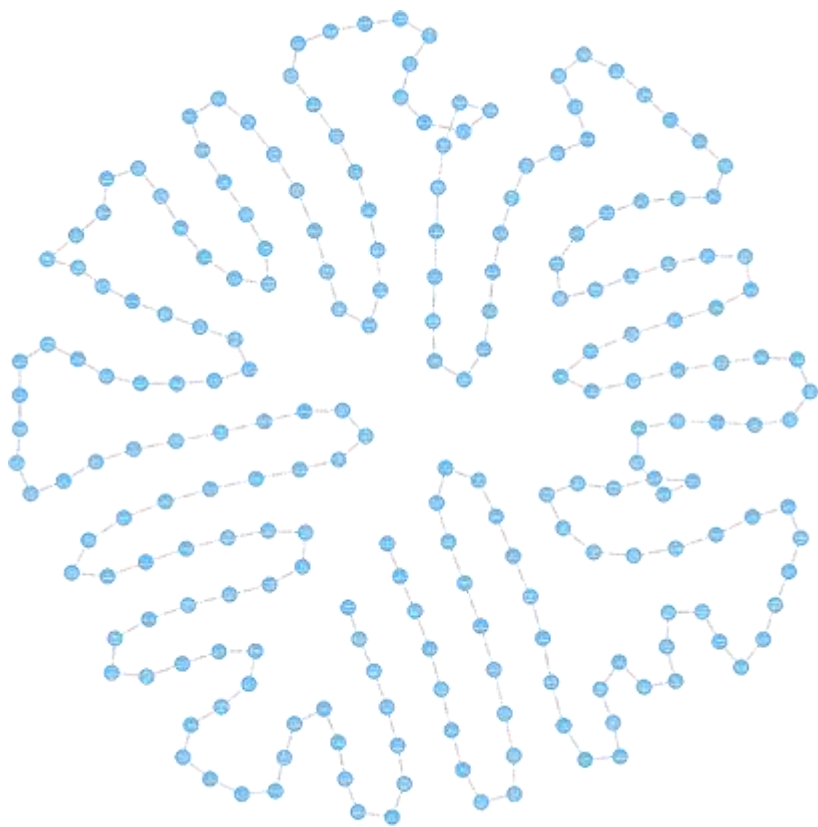
CaseId	Activity	Start_Timestamp	End_Timestamp
PP8	Send material	15-08-17 10:20	15-08-17 15:20
PP8	Confirm material arrival	15-08-17 15:20	16-08-17 07:19
PP8	Invisible task 2	16-08-17 07:19	16-08-17 07:20
PP8	Take material	16-08-17 07:20	16-08-17 07:29
PP8	Invisible task 3	16-08-17 07:29	16-08-17 07:30
PP8	Use material	16-08-17 07:30	23-08-17 07:29
PP8	Invisible task 4	23-08-17 07:29	23-08-17 07:30
PP8	Return material before due time	23-08-17 07:30	23-08-17 07:39
PP8	Invisible task 5	23-08-17 07:39	23-08-17 07:40
PP8	Send back material	23-08-17 07:40	23-08-17 12:40
PP8	Confirm material send back arrival	23-08-17 12:40	23-08-17 12:44
PP8	Invisible task 6	23-08-17 12:44	23-08-17 12:45
PP8	Check material condition	23-08-17 12:45	23-08-17 14:59
PP8	Invisible task 23	23-08-17 14:59	23-08-17 15:00
PP8	Confirm material in bad condition	23-08-17 15:00	23-08-17 15:04
PP8	Invisible task 24	23-08-17 15:04	23-08-17 15:05
PP8	Determine the amounts of fines to be paid	23-08-17 15:05	23-08-17 15:15
PP8	Send detail of the fines	23-08-17 15:15	23-08-17 15:25
PP8	Ask user to pay the fine	23-08-17 15:25	23-08-17 15:30
PP8	Pay the fines and confirm payment	23-08-17 15:30	24-08-17 07:30
PP8	Confirm payment	24-08-17 07:30	24-08-17 07:39
PP8	Invisible task 22	24-08-17 07:39	24-08-17 07:40
PP8	Confirm loan finish	24-08-17 07:40	24-08-17 07:45
PP9	Request material	25-08-17 07:05	25-08-17 07:10
PP9	Check availability	25-08-17 07:10	25-08-17 10:14
PP9	Invisible task 1	25-08-17 10:14	25-08-17 10:15
PP9	Confirm material availability	25-08-17 10:15	25-08-17 10:20
PP9	Send material	25-08-17 10:20	25-08-17 13:20
PP9	Confirm material arrival	25-08-17 13:20	25-08-17 13:24
PP9	Invisible task 2	25-08-17 13:24	25-08-17 13:25
PP9	Take material	25-08-17 13:25	25-08-17 13:39
PP9	Invisible task 3	25-08-17 13:39	25-08-17 13:40
PP9	Use material	25-08-17 13:40	11-09-17 07:39
PP9	Invisible task 19	11-09-17 07:39	11-09-17 07:40
PP9	Return material after due time	11-09-17 07:40	11-09-17 07:49
PP9	Invisible task 20	11-09-17 07:49	11-09-17 07:50
PP9	Send back material	11-09-17 07:50	11-09-17 11:00
PP9	Confirm material send back arrival	11-09-17 11:00	11-09-17 11:04
PP9	Invisible task 6	11-09-17 11:04	11-09-17 11:05
PP9	Check material condition	11-09-17 11:05	11-09-17 14:05
PP9	Invisible task 23	11-09-17 14:05	11-09-17 14:05
PP9	Confirm material in bad condition	11-09-17 14:05	11-09-17 14:09
PP9	Invisible task 24	11-09-17 14:09	11-09-17 14:10

CaseId	Activity	Start_Timestamp	End_Timestamp
PP9	Determine the amounts of fines to be paid	11-09-17 14:10	11-09-17 14:20
PP9	Send detail of the fines	11-09-17 14:20	11-09-17 14:30
PP9	Ask user to pay the fine	11-09-17 14:30	11-09-17 14:35
PP9	Pay the fines and confirm payment	11-09-17 14:35	11-09-17 17:40
PP9	Confirm payment	11-09-17 17:40	12-09-17 07:29
PP9	Invisible task 22	12-09-17 07:29	12-09-17 07:30
PP9	Confirm loan finish	12-09-17 07:30	12-09-17 07:35
PP10	Request material	13-09-17 07:05	13-09-17 07:10
PP10	Check availability	13-09-17 07:10	13-09-17 10:14
PP10	Invisible task 1	13-09-17 10:14	13-09-17 10:15
PP10	Confirm material availability	13-09-17 10:15	13-09-17 10:20
PP10	Send material	13-09-17 10:20	13-09-17 15:20
PP10	Confirm material arrival	13-09-17 15:20	14-09-17 07:09
PP10	Invisible task 2	14-09-17 07:09	14-09-17 07:10
PP10	Take material	14-09-17 07:10	14-09-17 07:19
PP10	Invisible task 3	14-09-17 07:19	14-09-17 07:20
PP10	Use material	14-09-17 07:20	25-09-17 07:19
PP10	Invisible task 11	25-09-17 07:19	25-09-17 07:20
PP10	Request lending extension	25-09-17 07:20	25-09-17 07:30
PP10	Ask lending extension availability	25-09-17 07:30	25-09-17 07:39
PP10	Invisible task 12	25-09-17 07:39	25-09-17 07:40
PP10	Allow lending extension	25-09-17 07:40	25-09-17 08:19
PP10	Invisible task 13	25-09-17 08:19	25-09-17 08:20
PP10	Use material	25-09-17 08:20	10-10-17 07:19
PP10	Invisible task 19	10-10-17 07:19	10-10-17 07:20
PP10	Return material after due time	10-10-17 07:20	10-10-17 07:29
PP10	Invisible task 20	10-10-17 07:29	10-10-17 07:30
PP10	Send back material	10-10-17 07:30	10-10-17 12:30
PP10	Confirm material send back arrival	10-10-17 12:30	10-10-17 12:34
PP10	Invisible task 6	10-10-17 12:34	10-10-17 12:35
PP10	Check material condition	10-10-17 12:35	10-10-17 14:49
PP10	Invisible task 7	10-10-17 14:49	10-10-17 14:50
PP10	Confirm material in good condition	10-10-17 14:50	10-10-17 14:54
PP10	Invisible task 21	10-10-17 14:54	10-10-17 14:55
PP10	Determine the amounts of fines to be paid	10-10-17 14:55	10-10-17 15:10
PP10	Send detail of the fines	10-10-17 15:10	10-10-17 15:20
PP10	Ask user to pay the fine	10-10-17 15:20	10-10-17 15:25
PP10	Pay the fines and confirm payment	10-10-17 15:25	10-10-17 20:25
PP10	Confirm payment	10-10-17 20:25	11-10-17 07:34
PP10	Invisible task 22	11-10-17 07:34	11-10-17 07:35
PP10	Confirm loan finish	11-10-17 07:35	11-10-17 07:40
PP11	Request material	11-10-17 07:40	12-10-17 07:10
PP11	Check availability	12-10-17 07:10	12-10-17 10:14

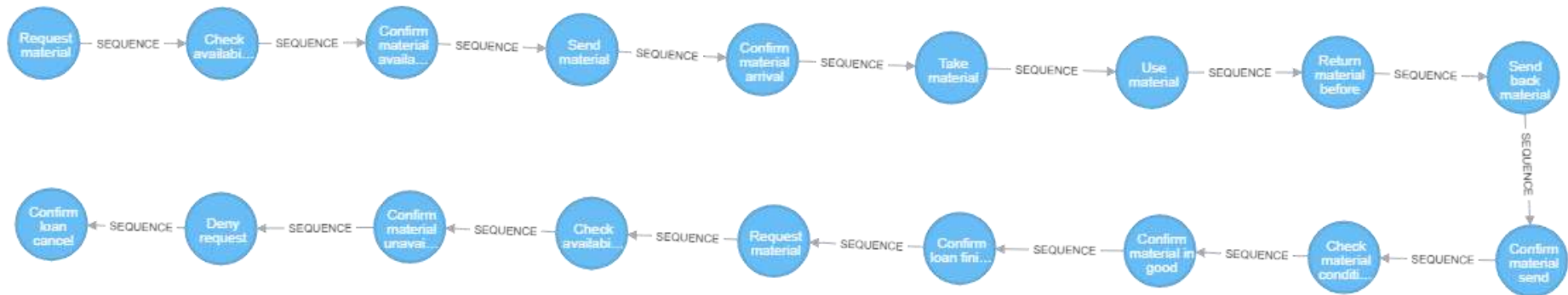
CaseId	Activity	Start_Timestamp	End_Timestamp
PP11	Invisible task 1	12-10-17 10:14	12-10-17 10:15
PP11	Confirm material availability	12-10-17 10:15	12-10-17 10:20
PP11	Send material	12-10-17 10:20	12-10-17 14:20
PP11	Confirm material arrival	12-10-17 14:20	12-10-17 14:24
PP11	Invisible task 2	12-10-17 14:24	12-10-17 14:25
PP11	Take material	12-10-17 14:25	12-10-17 14:39
PP11	Invisible task 3	12-10-17 14:39	12-10-17 14:40
PP11	Use material	12-10-17 14:40	23-10-17 07:49
PP11	Invisible task 11	23-10-17 07:49	23-10-17 07:50
PP11	Request lending extension	23-10-17 07:50	23-10-17 08:10
PP11	Ask lending extension availability	23-10-17 08:10	23-10-17 08:19
PP11	Invisible task 12	23-10-17 08:19	23-10-17 08:20
PP11	Allow lending extension	23-10-17 08:20	23-10-17 08:29
PP11	Invisible task 13	23-10-17 08:29	23-10-17 08:30
PP11	Use material	23-10-17 08:30	30-10-17 08:29
PP11	Invisible task 4	30-10-17 08:29	30-10-17 08:30
PP11	Return material before due time	30-10-17 08:30	30-10-17 08:39
PP11	Invisible task 5	30-10-17 08:39	30-10-17 08:40
PP11	Send back material	30-10-17 08:40	30-10-17 12:40
PP11	Confirm material send back arrival	30-10-17 12:40	30-10-17 12:44
PP11	Invisible task 6	30-10-17 12:44	30-10-17 12:45
PP11	Check material condition	30-10-17 12:45	30-10-17 14:44
PP11	Invisible task 23	30-10-17 14:44	30-10-17 14:45
PP11	Confirm material in bad condition	30-10-17 14:45	30-10-17 14:49
PP11	Invisible task 24	30-10-17 14:49	30-10-17 14:50
PP11	Determine the amounts of fines to be paid	30-10-17 14:50	30-10-17 15:05
PP11	Send detail of the fines	30-10-17 15:05	30-10-17 15:15
PP11	Ask user to pay the fine	30-10-17 15:15	30-10-17 15:20
PP11	Pay the fines and confirm payment	30-10-17 15:20	30-10-17 19:20
PP11	Confirm payment	30-10-17 19:20	31-10-17 07:34
PP11	Invisible task 22	31-10-17 07:34	31-10-17 07:35
PP11	Confirm loan finish	31-10-17 07:35	31-10-17 07:40
PP12	Request material	01-11-17 07:05	01-11-17 07:10
PP12	Check availability	01-11-17 07:10	01-11-17 10:14
PP12	Invisible task 1	01-11-17 10:14	01-11-17 10:15
PP12	Confirm material availability	01-11-17 10:15	01-11-17 10:20
PP12	Send material	01-11-17 10:20	01-11-17 13:50
PP12	Confirm material arrival	01-11-17 13:50	01-11-17 13:54
PP12	Invisible task 2	01-11-17 13:54	01-11-17 13:55
PP12	Take material	01-11-17 13:55	01-11-17 14:04
PP12	Invisible task 3	01-11-17 14:04	01-11-17 14:05
PP12	Use material	01-11-17 14:05	13-11-17 14:04
PP12	Invisible task 11	13-11-17 14:04	13-11-17 14:05
PP12	Request lending extension	13-11-17 14:05	13-11-17 14:15

CaseId	Activity	Start_Timestamp	End_Timestamp
PP12	Ask lending extension availability	13-11-17 14:15	13-11-17 14:24
PP12	Invisible task 14	13-11-17 14:24	13-11-17 14:25
PP12	Deny lending extension	13-11-17 14:25	13-11-17 14:49
PP12	Invisible task 15	13-11-17 14:49	13-11-17 14:50
PP12	Return material before due time	13-11-17 14:50	14-11-17 06:59
PP12	Invisible task 5	14-11-17 06:59	14-11-17 07:00
PP12	Send back material	14-11-17 07:00	14-11-17 10:00
PP12	Confirm material send back arrival	14-11-17 10:00	14-11-17 10:04
PP12	Invisible task 6	14-11-17 10:04	14-11-17 10:05
PP12	Check material condition	14-11-17 10:05	14-11-17 13:04
PP12	Invisible task 23	14-11-17 13:04	14-11-17 13:05
PP12	Confirm material in bad condition	14-11-17 13:05	14-11-17 13:09
PP12	Invisible task 24	14-11-17 13:09	14-11-17 13:10
PP12	Determine the amounts of fines to be paid	14-11-17 13:10	14-11-17 13:20
PP12	Send detail of the fines	14-11-17 13:20	14-11-17 13:25
PP12	Ask user to pay the fine	14-11-17 13:25	14-11-17 13:30
PP12	Pay the fines and confirm payment	14-11-17 13:30	14-11-17 15:30
PP12	Confirm payment	14-11-17 15:30	15-11-17 07:09
PP12	Invisible task 22	15-11-17 07:09	15-11-17 07:10
PP12	Confirm loan finish	15-11-17 07:10	15-11-17 07:15
PP13	Request material	29-11-17 07:15	29-11-17 07:20
PP13	Check availability	29-11-17 07:20	30-11-17 10:24
PP13	Invisible task 1	30-11-17 10:24	30-11-17 10:25
PP13	Confirm material availability	30-11-17 10:25	30-11-17 10:30
PP13	Send material	30-11-17 10:30	30-11-17 14:30
PP13	Confirm material arrival	30-11-17 14:30	30-11-17 14:34
PP13	Invisible task 2	30-11-17 14:34	30-11-17 14:35
PP13	Take material	30-11-17 14:35	30-11-17 14:34
PP13	Invisible task 3	30-11-17 14:34	30-11-17 14:35
PP13	Use material	30-11-17 14:35	10-12-17 07:19
PP13	Invisible task 11	10-12-17 07:19	10-12-17 07:20
PP13	Request lending extension	10-12-17 07:20	10-12-17 07:30
PP13	Ask lending extension availability	10-12-17 07:30	10-12-17 07:34
PP13	Invisible task 12	10-12-17 07:34	10-12-17 07:35
PP13	Allow lending extension	10-12-17 07:35	10-12-17 08:34
PP13	Invisible task 13	10-12-17 08:34	10-12-17 08:35
PP13	Use material	10-12-17 08:35	21-12-17 08:34
PP13	Invisible task 11	21-12-17 08:34	21-12-17 08:35
PP13	Request lending extension	21-12-17 08:35	21-12-17 08:45
PP13	Ask lending extension availability	21-12-17 08:45	21-12-17 08:49
PP13	Invisible task 12	21-12-17 08:49	21-12-17 08:50
PP13	Allow lending extension	21-12-17 08:50	21-12-17 09:49
PP13	Invisible task 13	21-12-17 09:49	21-12-17 09:50
PP13	Use material	21-12-17 09:50	03-01-18 07:49

CaseId	Activity	Start_Timestamp	End_Timestamp
PP13	Invisible task 15	03-01-18 07:49	03-01-18 07:50
PP13	Return material before due time	03-01-18 07:50	03-01-18 07:59
PP13	Invisible task 5	03-01-18 07:59	03-01-18 08:00
PP13	Send back material	03-01-18 08:00	03-01-18 11:30
PP13	Confirm material send back arrival	03-01-18 11:30	03-01-18 11:34
PP13	Invisible task 6	03-01-18 11:34	03-01-18 11:35
PP13	Check material condition	03-01-18 11:35	03-01-18 14:34
PP13	Invisible task 7	03-01-18 14:34	03-01-18 14:35
PP13	Confirm material in good condition	03-01-18 14:35	03-01-18 14:39
PP13	Invisible task 8	03-01-18 14:39	03-01-18 14:40
PP13	Confirm loan finish	03-01-18 14:40	03-01-18 14:45

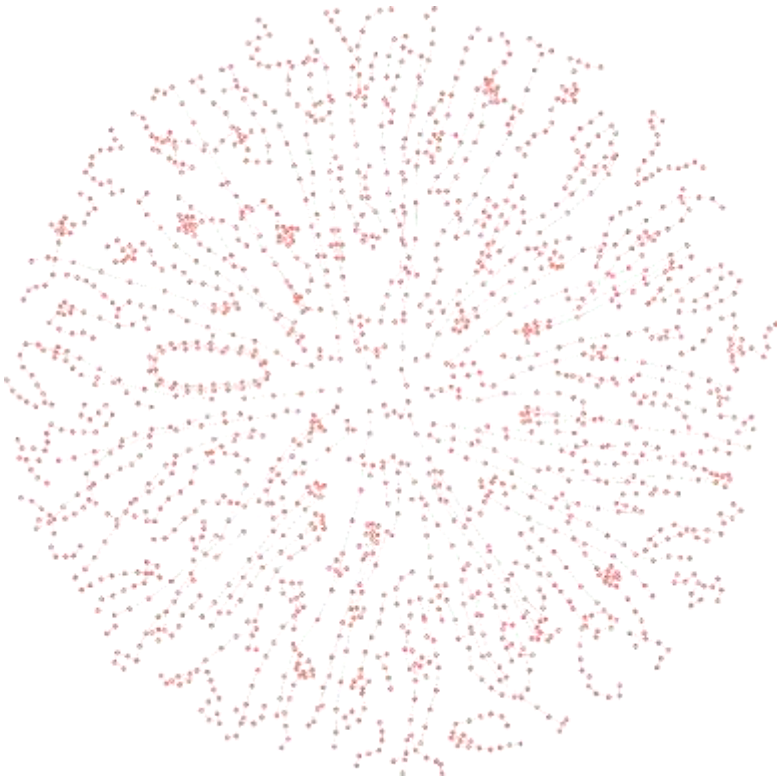


Gambar 7.1 Model Proses Bisnis Sequence



Gambar 7.2 Model Prroses Bisnis *Sequence* dengan CaseId PP1 dan PP2

[Halaman ini sengaja dikosongkan]



Gambar 7.3 Model Proses Bisnis Penemuan *Sequence Case*

[Halaman ini sengaja dikosongkan]

DAFTAR ISTILAH

AND	: relasi yang menunjukkan bahwa seluruh aktivitas pada relasi ini harus dijalankan semua di tiap proses.
<i>Case</i>	: Suatu kasus tertentu pada <i>event log</i> . Kasus tertentu tersebut dapat berupa suatu kasus dalam memproduksi suatu barang tertentu, karena <i>event log</i> dapat terdiri dari catatan dari proses eksekusi pembuatan banyak barang atau proses eksekusi dari banyak kasus proses.
<i>Case ID</i>	: Nomor identitas dari kasus tertentu pada <i>event log</i> .
<i>Event</i>	: Aktivitas
<i>Event Log</i>	: Suatu kumpulan eksekusi proses berdasarkan data aktivitas proses bisnis yang disimpan dalam bentuk tertentu.
<i>Node</i>	: unit dalam graf
OR	: relasi yang menunjukkan bahwa seluruh aktivitas pada relasi tersebut dapat dijalankan semua atau dipilih beberapa untuk tiap proses
<i>Process Discovery</i>	: pembentukan model proses
<i>Process Mining</i>	: penggalian proses
<i>Skip activity</i>	: aktivitas yang hilang
SOP	: <i>Standard Operating Procedure</i>
<i>Trace</i>	: rangkaian dari aktivitas.
XOR	: relasi yang menunjukkan bahwa seluruh aktivitas pada relasi ini harus dipilih salah satu untuk dijalankan

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Hendra Darmawan, lahir pada 29 Mei 1996 di Jember. Penulis merupakan seorang mahasiswa yang sedang menempuh studi di Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember (ITS). Selama menempuh pendidikan di kampus, penulis juga aktif di organisasi kemahasiswaan, antara lain Staff Kaderisasi dan Pemetaan Himpunan Mahasiswa Teknik Computer-Informatika (HMTTC), Koordinator 1 Steering Committee Himpunan Mahasiswa Teknik Computer-Informatika, dan Staff National Seminar of Technology (NST) Schematics ITS.